ESD-TR-75-66

MTR-2925

PRELIMINARY AFBITS NETWORK CONTROL SYSTEM

JUNE 1975

Prepared for

DEPUTY FOR DEVELOPMENT PLANS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Bedford, Massachusetts

Project No. 7040
Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract No. F19628-75-C-0001

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

ALPH L. WESTLEY/GS-13
Project Manager

FOR THE COMMANDER

CARL A. SEGERSTROM/GS-15
Acting Director, Technology
Deputy for Development Plans

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-75-66	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) PRELIMINARY AFBITS NETWORK CONTROL SYSTEM		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER MTR-2925
7. AUTHOR(s) H.R. Clark, M.S. Cogan, H.I. Cohen, G.C. Dempsey, R.P. Witt, I.H. Ziemba		8. CONTRACT OR GRANT NUMBER(s) F19628-75-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS The MITRE Corporation Box 208 Bedford, MA 01730		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 7040
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Development Plans Electronic Systems Division, AFSC Hanscom Air Force Base, Bedford, MA 01731		12. REPORT DATE JUNE 1975 13. NUMBER OF PAGES 175
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

BROADBAND COMMUNICATIONS
COAXIAL CABLE
MINICOMPUTERS
NETWORK CONTROL
SUPERVISION AND SIGNALING
TIME DIVISION MULTIPLEX

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a network control system for a coaxial cable based, multi-mode communication facility to be utilized for audio, video and data services on future Air Force Base communications installations. A detailed description is provided of the software for a minicomputer-based network control program and the hardware for a low cost out-of-band signaling and supervision system that can be frequency multiplexed on a broadband communications system.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

LIST OF FIGURES AND TABLES

LIST OF FIGURES AND TABLES (Continued)

LIST OF FIGURES AND TABLES (Concluded)

GLOSSARY OF ABBREVIATIONS AND ACRONYMS

AFBITS	Air Force Base Information Transfer System
ASCII	American Standard Code for Information Interchange
ATP	Automated Telecommunications Program
B.W.C.	Buffer Word Count
CAPS 11	DEC Cassette operating system for PDP-11
CATV	Community Antenna Television
CCTV	Closed Circuit Television
CP	Central Processor
DISP	System Dispatcher
DST	Downstream State Timer
EQTBL	Common Equipment Tables
ESC	Escape
FDM	Frequency Division Multiplex
FIFO	First-In/First-Out
FLM	FIFO Level Monitor
HPAC	Hardware Poll Address Counter
HIC	Headend Interface Controller
INIT	Initialize
IOM	Input/Output Monitor
IOXLPT	Input/Output Executive-Line Printer
IPRT	Inter-matrix Path Route Table
MOS	Metalization of Oxide on Silicon
NCP	Network Control Processor
NETCOP	Network Control Program
ODT-11X	On-Line Debugging Technique for PDP-11
RTI	Return from Interrupt
RTS	Return from Subroutine
SASI	Signaling and Supervision Input
SASO	Signaling and Supervision Output
SCR	Status and Control Register
SCU	Switch Control Unit
S&S	Signaling and Supervision
SPFT	Service Package Format Table
SUSM	Set Up Switch Messages
TCS	Total Communication System
TDM	Time Division Multiplex
TGST	Trunk Group Status Table
UART	Universal Asynchronous Receiver/Transmitter
UDTL	User Data Tables
USCR	Upstream Status Control Word
UST	Upstream State Timer
VDT	Visual Display Terminal

SECTION I

1.0 INTRODUCTION

1.1 AFBITS Concept

The Air Force Base Information Transfer System (AFBITS) is an integrated multimode communications system concept designed to satisfy existing and future needs for information transfer on Air Force bases. Through use of broadband transmission media such as coaxial cable a multimode information transfer system can be implemented which provides audio, digital data, message, correspondence and video communications simultaneously in a variety of connectivity patterns. These various connectivities include: telephone point-to-point; the many-to-one connectivities for remote terminal access to computers; the one-to-many or broadcast mode; and the many-to-many or conferencing mode. Such communications capabilities can handle a variety of subscriber devices including visual display terminals which allow information transfer to be accomplished on a soft copy basis. With these capabilities, a number of simultaneous user services such as text processing, automated correspondence and message distribution, remote data input and access to computer and microfiche data bases, can be provided.

1.2 Background

The principal electrical communications facility on Air Force bases today is the dial telephone system which provides point-to-point voice services and some limited digital data transmission capability on either a switched or dedicated line basis. Limited video services are provided by CCTV facilities on some bases. The primary means of information transfer on Air Force bases however continues to be through physical transport of hard copy in the form of letters, memoranda, reports and cards or magnetic tapes.

The production and handling of the large amounts of paper necessary to satisfy information transfer requirements on Air Force bases requires substantial numbers of personnel to provide this service in the Communications, Administration and Data Automation functions. Additionally all mission and function users of the information transfer system spend an appreciable portion of their time in the handling of paper in order to accomplish their primary job function.

The implementation of a broadband integrated multi-mode telecommunications system such as the AFBITS would permit a substantial portion of this information transfer to be accomplished by means of automated electrical soft copy for the generation, composition, storage and transmission of information. The amount of

paper hard copy utilized for information transfer purposes therefore could be substantially reduced.

1.3 Technical Feasibility Demonstrations

In order to evaluate the technical feasibility of developing a broadband integrated multi-mode information transfer system like AFBITS an experimental evaluation facility consisting of representative types of data and video terminals and devices interconnected by broadband cable and switches has been configured. With this setup, a variety of video display and typewriter oriented terminals have been interfaced and interconnected through a broadband switch. Various types of video devices such as surveillance cameras, an automatic microfiche retrieval unit, framegrabbers, video tape recorders and copiers have been interconnected and evaluated (Reference 1). An interface has also been effected with a coaxial cable facility

The combining of a multiplicity of services and signals under a single transmission plant under the control of a single network control processor involves a complex supervision and signaling capability. This multimode control is required to handle time-division-multiplexing of both data and video (single addressed frame) signals and digitally controlled frequency-division selectable channel assignments and space-division switching of video signals. Connectivities will include: point-to-point services (as with the telephone), many-to-one connections (typical of remote computer terminals) and the more significant complexity of the "many-to-several" connection pattern. This latter pattern is exemplified by the need for many terminals to communicate interactively with a computer while other terminals are either operating simultaneously with the message center or are working with a microfiche data retrieval library or other centralized information handling services.

A network configuration and control system organization has been structured and programmed for the evaluation of the feasibility and complexity of multimode communication control. This configuration employs a Digital Equipment Corporation PDP-11 computer and an out-of-band signaling and supervision system multiplexed on coaxial cable. Because of the complexity of the control and the wide diversity of system equipments, this signaling and supervision subsystem has been designed as a time division multiplex (TDM) data distribution capability. The cost and complexity of the individual subscriber interface units has been minimized by the use of a polled asynchronous TDM mode of operation.

1.4 Operational Testbed

Although the implementation concepts for AFBITS do not require any technical breakthroughs, these concepts do incorporate a combination of previously independent technologies and involve design of new forms for well known equipments. Of more significance is the necessity to evaluate the AFBITS system, which will provide many new classes of service, in relation to user information transfer needs in an operational environment. The AFBITS testbed would provide the environment for the operational evaluation of soft copy information transfer services and permit evolutionary modification for an operational configuration.

The testbed design must stress versatility in order to accommodate both the evolutionary development and the ability to utilize and evaluate a variety of available services. In spite of the need for such adaptability, a number of interim standards are required to minimize interface compatibility and procurement problems. To this end, it is probable that standard coaxial cable distribution practice and equipments will be utilized wherever possible.

SECTION II

2.0 AFBITS SYSTEM DESIGN

2.1 AFBITS Configuration

The AFBITS will employ coaxial cable to provide a broadband transmission/distribution facility on an Air Force base. Through appropriate combinations of time division multiplexing (TDM) and frequency division multiplexing (FDM) techniques a capability will be provided for handling digital data, video and audio signals between thousands of subscriber terminals. The backbone trunking will be by means of a dual coaxial cable, one for downstream signals and another for upstream signals as shown in Figure 1. Individual subscriber connections may be either through branch drops from the main trunks or through switched hubs for dense subscriber concentrations.

These "hubs" serve as distribution points for groups of system users. At each "hub" video switching facilities under control of the centralized network control processor are provided. All nonvideo signals are routed directly to the subscriber terminals, by-passing the video switch. In addition to the switch, the "hub" may contain text editing minicomputers and shared-user equipments for page storage, character generation and cathode ray tube refresh as well as video framegrabbers which are used in connection with a remote access microfiche data retrieval service. The hub facilities provide for significant equipment and cost sharing. In addition, they make the accommodation of all system services feasible without the requirement for untenable trunk bandwidths.

Signal trunking between elements (terminals and/or switches) of the system can be accomplished by the use of a shielded coaxial cable. Such cables have been utilized extensively for CATV and provide a benign transmission environment. A large multiplicity of signals can be carried simultaneously within a 300 megahertz bandwidth on each cable. One cable will be utilized for transmissions "downstream" to the individual subscribers. The other cable will collect subscriber transmissions and route them "upstream" to a headend facility. At the headend facility, the "upstream" signals will generally be inserted onto the "downstream" channel. At this point, the upstream signals can be intercepted for off-base transmission through the AUTODIN interface at the message center or used by other than individual base subscribers. All signal amplitudes are carefully controlled so that the problem of

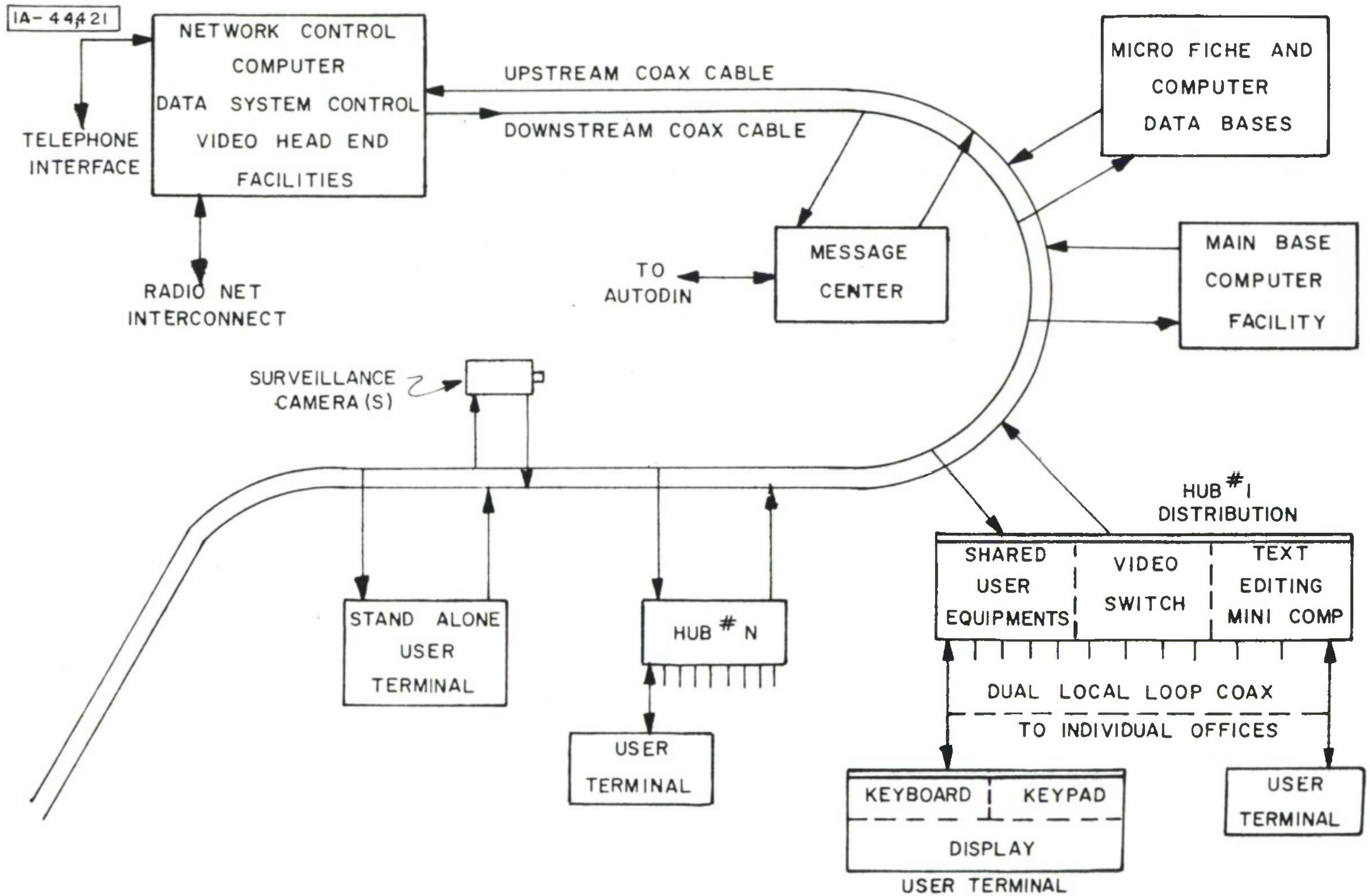


Figure 1 AFBITS SIGNAL DISTRIBUTION SYSTEM

adjacent channel interference is 60 to 80 db less serious than that encountered in conventional radio transmissions. The cable shielding eliminates the vast majority of external interference problems.

The limited geographical area of an Air Force base will permit all signals to be handled with less than ten cable repeaters. This small number of repeaters implies the ability to preserve a significantly better signal-to-noise ratio throughout the distribution system than is customarily encountered in CATV practice. The utilization of dual coaxial cables eliminates many of the serious intermodulation distortion problems encountered in single cable two-way transmission. For standard data services on the cable, MIL 188-100 has been selected. Other standards will, of course, be applied in specifying the pilot base/testbed implementation.

Cable attenuation will require the use of signal amplifiers (repeaters) at a spacing of every 1,000 to 2,000 feet, depending upon the type of cable chosen. Nonlinearities in these repeaters can create intermodulation distortion products which can, in turn, provide a low level signal interference. The use of dual coaxial cables greatly reduces this potential interference compared to what would exist on a single cable that has been subdivided into frequency bands for "upstream" and "downstream" transmissions. Conventional practice calls for the use of repeaters whenever the highest frequency signals have been attenuated by approximately 22 db.

A typical trunking frequency plan is depicted in Figure 2. Three note worthy features are evident in this frequency plan. No utilization of the first several megahertz of cable bandwidth is proposed because of high radio interference and impulse noise interferences. Secondly, data services (which are a potentially high source of interference) are grouped together at the lower end of the cable spectrum. The digital data signals are thus confined so that only rugged data services are exposed to their own potentially high interference. Thirdly, provisions are made to preserve the standard FCC frequency assignments for VHF TV and FM. The frequency allocations for telephone and for other video service systems, has not been definitized but is probably best handled with the lowest available frequency band provided for the switchless FDM telephone services.

Figure 3 illustrates a local-loop frequency plan which can be accommodated within a 60 MHz bandwidth coaxial cable subscriber loop

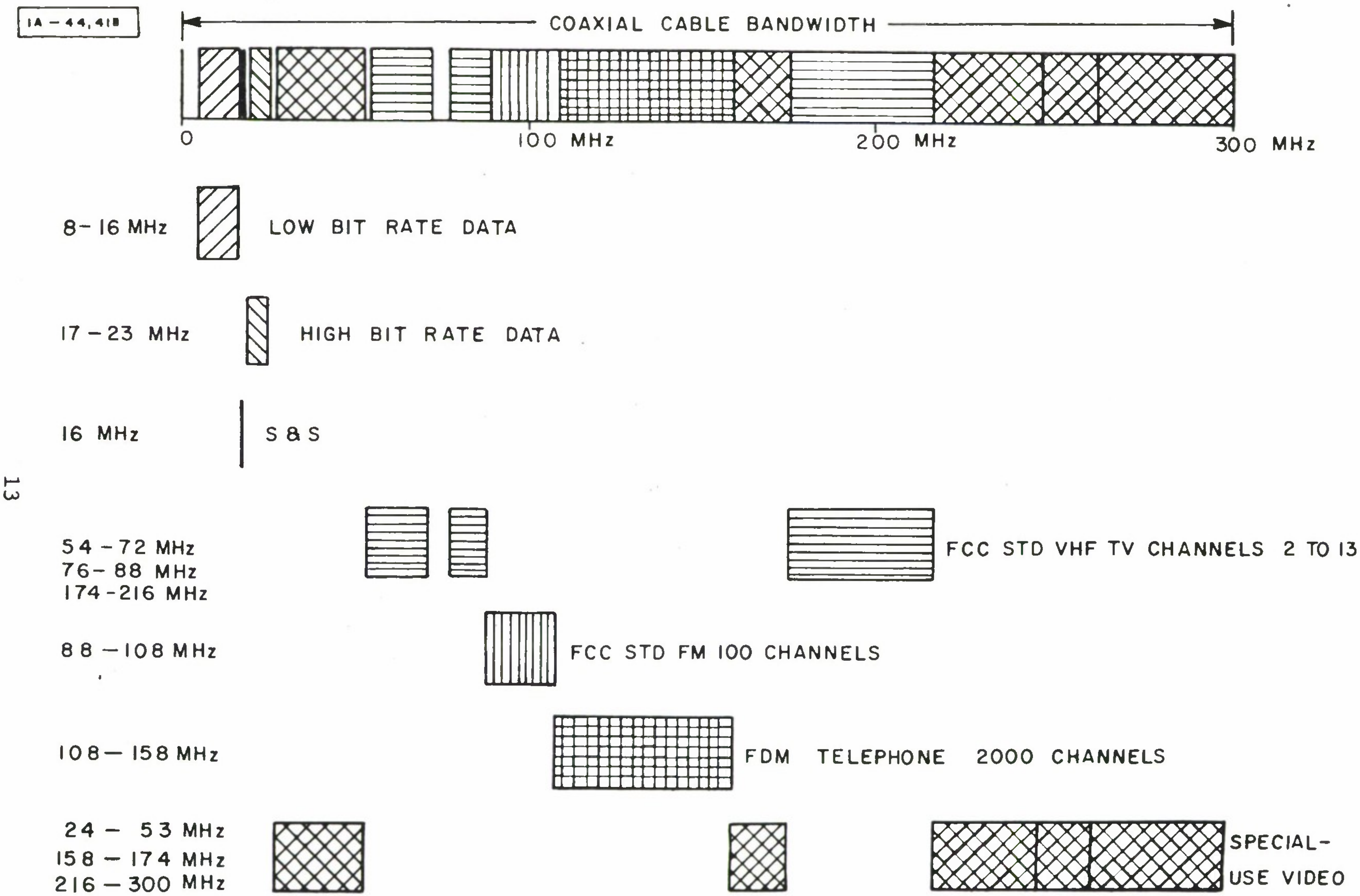


Figure 2 TRUNK FREQUENCY PLAN

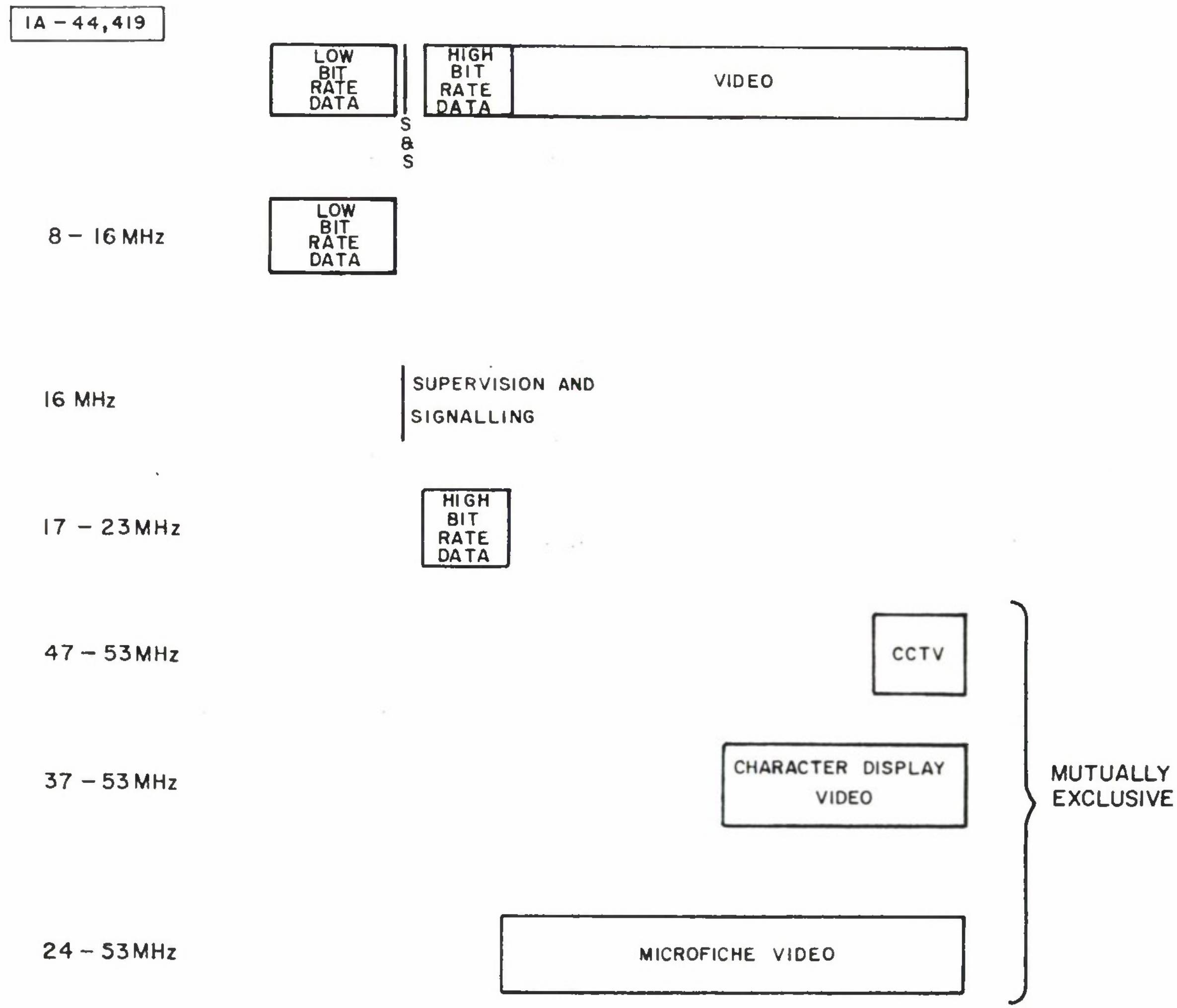


Figure 3. LOCAL LOOP FREQUENCY PLAN

and is quite similar to the lower end of the trunking frequency allocation. The signaling and supervision channel and the low bit rate data channel can, therefore, be passed directly from the trunk to subscribers and back again. Accommodation of local-loop video is through modulation into the next higher available frequencies. High-bit-rate data is placed between other data services and the video services. The final decisions on this allocation will be deferred pending further technical and system needs analysis.

2.2 AFBITS Control Concepts

The current AFBITS design provides six basic user connectivities plus the ability to add or delete any of these services from/to an already existent call. These services are:

- Standard point-to-point telephone call,
- Two-way interactive data (separate low speed and high speed facilities),
- One-way video (4 MHz bandwidth),
- Receive video, transmit data (4 MHz or 15 MHz bandwidth),
- Microfiche retrieval (video) and
- Three-way service (simultaneous audio, video and data).

The phone service provided on the AFBITS coaxial cable system is planned to utilize digitally addressable 25 kHz FDM channels. Thus, 2,000 simultaneous channels could be provided in a bandwidth equivalent to only 1/6 of the coaxial capacity. This appears to be an appropriate bandwidth allocation, considering the continued prime importance of the base telephone services.

Although the AFBITS concept provides for simultaneous utilization of audio, video and data facilities, it may be economically more expedient to integrate the telephone services indirectly. A subscriber call, requiring a telephone connection as well as the use of other system facilities, would be directed to the network control processor. This computer will then complete all of the call set up except for those activities connected with the telephone service. Furthermore, the network control processor will identify the connectivity required for the telephone service and

will transfer this information to a telephone switch controller. Thus, all aspects of the multiple communications required can be accomplished even while retaining a conventional common control type of telephone switch.

The digital data services provided will range from teletype equivalent speeds of 75 bps up to data rates useful for secure voice, high speed facsimile and computer-to-computer transfer (56 kbs). This capability can be economically provided if two separate classes of data services are utilized. All data transmission at average data rates of 2.4 kbs and below will be furnished on an asynchronous TDM channel. Each of the 500 data subscribers can be serviced within a 10 millisecond "frame" time by utilizing a nominal 4 mbs transmission rate. Only four characters of buffer storage are required and no special control for lower data rates (other than null codes) need be made. Higher bit rate data will be accommodated by setting up circuit-connected 100 kHz bandwidth FDM channels for high bit rate digital data services. Likewise, these same FDM channels could be used for bandwidth analog signals.

One-way video services will also be supplied by digitally selectable FDM channels. These services will be available for surveillance and possibly for monitoring TV type, education and information program materials.

Another basic video service requires 15 MHz bandwidth for a standard video display terminal capability where character-display sharpness requires the wide bandwidth. This capability is required for simple low cost Keyboard/Monitor terminals driven by remotely-located character generator/refresh electronics, probably located at the local hubs. In this case, the video must be passed from the remotely located character generator to a video display terminal which consists of not much more than a 15 MHz TV monitor (broadband) and keyboard. The call connection would be provided through one of several shareable character generator and display packages. A space-division video switch facility makes sharing possible.

A more complex call connectivity is required for microfiche retrieval. This service requires high resolution commensurate with a 30 MHz bandwidth. Normally, the general interest centralized automated microfiche files would be utilized, simultaneously, by many subscribers. Separate 30 MHz channels from the microfiche retrieval units are not feasible from a transmission viewpoint. In addition, it would not be desirable for one user to tie up the retrieval units continuously during his file perusal. The use of a microfiche retrieval system calls for a means of selecting

individual pages of stored material, relaying an image of this material to the user, and allowing the user to view this image for a relatively long period of time. This could be done by constant rescanning of the microfiche material, but that would involve the commitment of both the microfiche unit and the broadband video trunk channel to the user. Other users are, therefore, precluded from access to the microfiche equipment or the use of the video channel.

An alternate mode of operation would be to transfer the image to a storage mechanism (framegrabber) that is shareable, and is located in close proximity to the user. As soon as the transfer is complete, both the broadband communication channel and the microfiche retrieval unit can be made available to other subscribers. The local storage unit can then be connected to the requesting user's local loop for repetitive viewing on his scanner at standard 30 frame/sec. flicker free rates. The system needs are accommodated by passing individually addressed video frames, (or short multiple-frame transmission) through the 30 MHz transmission channel and onto frame storage equipment at the user's hub. In this way, common use equipment is tied up for only a few seconds. Additionally, as little as one-thirtieth of a second of trunk tie-up is required for each page retrieved. This mode of operation would be consistent with the provision of many microfiche retrieval units with a large library file. Each such unit would access only a portion of the library file.

Simultaneous audio, video, and data connectivities are also provided as a standard capability. This service would couple a two-way telephone conversation to the simultaneous (both parties) video viewing of computerlike interactive data or microfiche material. The originating subscriber could also have a data connection to the common information source.

Complete versatility for audio, video, and data services requires more than the above described capabilities. Additions and/or deletions to these capabilities will be defined in terms of these basic services. For example, the basic microfiche retrieval capability may be supplemented by a simultaneous telephone call or the basic surveillance capability may be supplemented by a parallel transfer of the video to another monitor and by initiating a phone call for conferencing.

2.3 TDM Signaling and Supervision Facility

The control of the diverse interconnections in the AFBITS multimode communication complex requires the incorporation of a

complex signaling and supervision (S&S) capability. In order to provide a capability that encompasses telephone, data, and a variety of video connectivities an out-of-band signaling technique is employed. This separation of signals permits easier implementation and is especially needed to permit call modification after an initial setup.

The signaling and supervision (S&S) facility initially implemented consists of user keypads, asynchronous TDM transmission equipments, a network control processor and ancillary support equipments as shown in Figure 4. For simplicity and economy, the transmission doctrine selected was a polled asynchronous TDM scheme. A systematic polling of subscribers is accomplished at a high bit rate by the successive transmission of all addresses in the system. A time gap (several microseconds) is provided after each polling address. This gap may contain supervisory information. As soon as a polled terminal recognizes its identifier, a transmission sequence is initiated. After a short, but uniquely specified delay (a function of the transmission distance to the network controller), previously stored Keypad digits may be transmitted through the upstream channel.

At the computer, the Keypad digits are collected and entered on a program-interrupt basis. The computer will store such input information along with the subscriber-code identifier until a complete sequence is received. The sequence length is a function of the nature of the service requested. Therefore, a call for microfiche retrieval will initiate queries through the S&S channel to determine equipment availability and will also initiate commands to set up the desired connectivity. All such queries and commands are transmitted with a fixed-message format as defined in Section III. The message bits are utilized to specify the request. System equipments, such as switches, as well as subscriber terminals, utilize the same type of address detection and message storage facility. Downstream messages from the network control processor are inserted into the polling (and keypad response) sequence as required.

Instructions from the computer to set up a data channel (duplex) on the asynchronous TDM data channel begin by polling the destination-subscriber terminal for its status. A return message would indicate status information, including facts like power-off, busy, ready-to-receive, etc. The computer would operate on or would retransmit elements of this data to the initiating subscriber. An indirect addressing technique is planned for the system wherein directory addresses would not correspond to the polling addresses

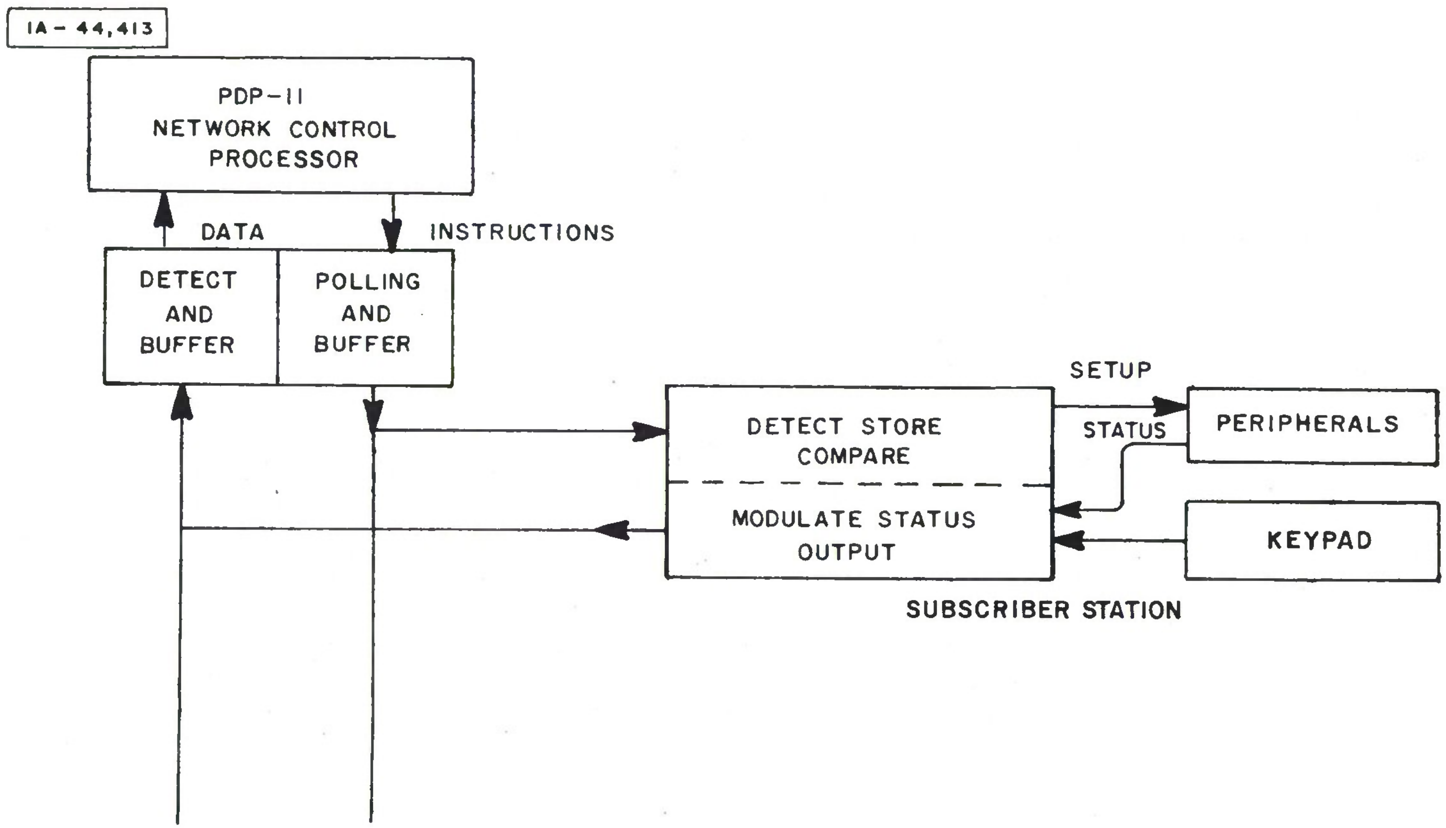


Figure 4 SIGNALING AND SUPERVISION SYSTEM

used to identify a subscriber terminal. Indirect addressing prevents direct subscriber access to terminal addresses and also permits user accountability. Because of indirect addressing, the computer must then transmit the address to be utilized (separately) to initiator and destination terminals. Within these terminals, the subscriber equipment will insert the destination address to be used as the header for all asynchronous TDM messages. Likewise, the receiver terminal equipment will receive and utilize the initiator's address for any return messages.

In a similar manner, the network control processor will determine and effect those operations needed to set up digitally addressable FDM frequency synthesizers for voice or video channels. Furthermore, the necessary video switch cross-point closures or shared-equipment facilities can be connected to complete the desired call. Shared-equipments will include character-generator refresh equipments, framegrabbers, microfiche facilities, etc.

As previously described, any call in progress may be modified by the addition or deletion of elements that are themselves one of the basic system services. A further system capability consists of the terminal's ability to self-initiate status messages while calls are in progress. Analysis of traffic flow and computer load indicates that this would be highly undesirable from idle terminals, but may prove necessary for some busy terminal applications. The provision of this capability as a part of the S&S facility would require some additional control logic at each terminal location.

A full size base capacity of 16,000 subscribers could be accommodated on a 100 millisecond polling sequence with a 10 megabit per second transmission rate. Alternative schemes, utilizing polling restricted to subscribers that have been identified (flagged) for services, are also available at significant bit-rate and bandwidth savings. The S&S design requirement is that support of subscriber services should not be apparent to the subscriber (user). The key factors are ease of use and the absence of long delays while creating connectivity. Relaxation of the burden of input-message processing by the software is accomplished by scanning each possible message for valid or new data before computer entry.

2.4 TDM Digital Data Service

In AFBITS, large numbers of subscriber terminals must be able to send and receive digital data to and from other terminals, computer data bases, and various peripheral devices. Such digital data transfer can be provided on a cable system by means of a Time

Division Multiplex (TDM) system in which a high-speed digital data stream is divided into time slots. This provides essentially simultaneous digital data service to a large number of individual subscribers.

In a normal synchronous TDM transmission system, each time slot is assigned to a given subscriber and the movement of data from one time slot in a frame to another time slot provides for data transfer between links of a network. Generally speaking, synchronous systems require the user equipment to be matched in speed to the occurrence of the time slots. Each terminal must be able to recognize the address of each time slot or else system demultiplexing equipments must be provided. Recently, concepts of packet communications have been developed in which short (even partial) messages and addresses are routed through a network by an originating subscriber to the appropriate destination. A packet communication system can be viewed as a dynamically reassignable TDM system with temporary message storage at network nodes.

By contrast, the asynchronous TDM system may be viewed as the equivalent of providing independent circuit-switched facilities between users, with no requirements for precise synchronization or local storage of data. Additionally, no requirement for complex traffic control exists in an asynchronous TDM system since each terminal responds only when polled.

The general method of operation for an asynchronous polled TDM data service is as follows. The TDM polling controller, which may be centrally located at the systems controller location, transmits polling messages in the time slot assigned for this service on the downstream cable. The format of these polling messages, as shown in Figure 5, consists of the "poll" that is the 14 bit address of the terminal being polled. Whenever a terminal recognizes its address, it responds on the upstream cable in the time slot assigned for this service with any data message it may have to transmit. The format of the upstream data messages, as shown in Figure 5, consists of the 14 bit address of the terminal for whom the message is intended and is followed by a message of four 8-bit characters. The TDM polling controller also serves the function of transmitting each message received on the upstream cable to the downstream cable. The downstream messages are inserted in the space between the polling messages.

In a typical operational system, a polling rate of 50,000 polls per second would permit 500 terminals to be polled every hundredth of a second. Since each terminal can transmit a total of four 8-bit

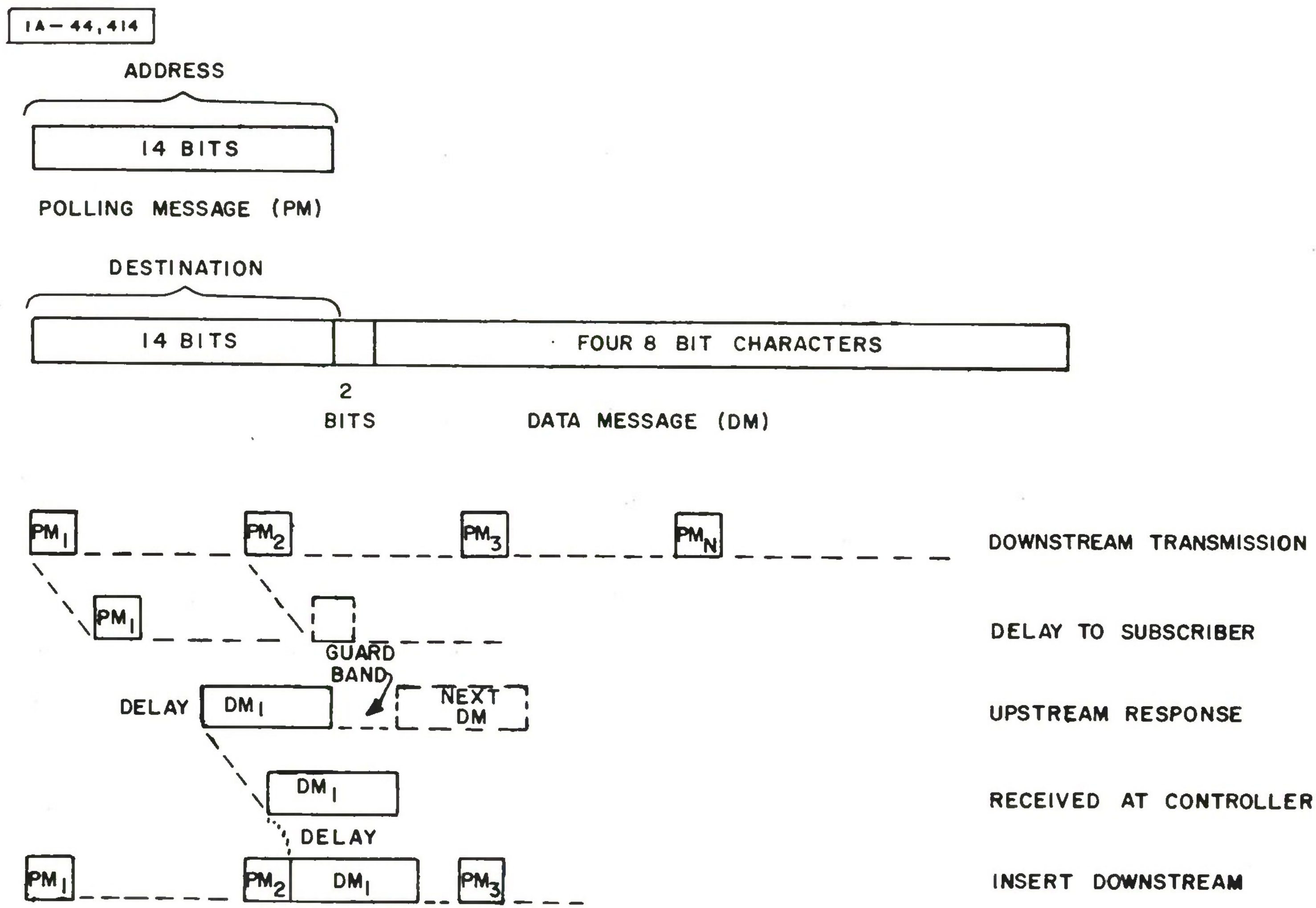


Figure 5 COMMON USER DATA MESSAGE FORMAT

characters on each poll the average information transfer rate per terminal is in excess of 2400 bits/second.

Within this system, fixed time slots are assigned to each subscriber on a data bus. These time slots reoccur at rates sufficiently fast enough to permit more than 2.4 kbs transmissions per subscriber with only four characters of storage between intervals. The actual transmission carries with it the intended receiver's address and will occur on a character basis with start, parity, and stop bits added to the eight information bits. Transmission at data rates less than the maximum is accommodated by inserting null characters. Each accepted message need not be rigidly tied to any system operating frequency. Guard bands (in time) are provided between the end of one transmission and initiation of the next. These guard bands, along with start, stop, and parity bits, nearly double the transmission rate to 4 megabaud to accommodate 500 users simultaneously.

SECTION III

3.0 AFBITS SIGNALING AND SUPERVISION SUBSYSTEM

3.1 Basic Concepts

As discussed earlier, AFBITS uses an out-of-band Supervision and Signaling (S&S) subsystem to perform control and management of devices and their connectivities. The use of out-of-band signaling permits these control and management functions to be requested and implemented independent of the subscriber's activity, or of the status of the work station equipment.

The S&S subsystem contains several major functional elements including the Network Control Processor (NCP) and its operational software, the interfacing head-end controller, and keypads that provide the physical and operational link to both the subscriber-user, and to other devices in the system.

The S&S functions by means of a polled asynchronous time division multiplex scheme wherein sequential polling is done to interrogate each element, and all transmissions between the headend controller to a particular element are handled asynchronously.

This section treats the design characteristics of the keypads that are used at the subscriber work station and the head-end controller that is the vehicle to interface the keypad signals to the Network Control Processor.

The performance requirements that S&S must meet are summed up in the following basic functions.

- Handle requests for connectivity,
- Authenticate request & requestor,
- Ascertain availability,
- Establish connectivity,
- Monitor connectivity and
- Report status.

3.2 Signaling and Supervision Subsystem Components

3.2.1 Major Operating S&S Subsystem Components

The major operating components of the S&S subsystem are shown in Figure 6. The NCP controls the Headend Interface Controller through the PDP-11 UNIBUS. The keypad signals into and out of this Headend Interface are UP and DOWNSTREAM DATA, to and from the headend pilot oscillator and modem. The modem services the cable, both UP and DOWN respectively. At each subscriber station, demodulation of the data and recovery of the pilot frequency takes place. The pilot frequency provides the clock that ensures that the data clock stays synchronized with the data, permitting the subscriber station keypad to receive the poll.

3.2.2 S&S Signal Paths

The S&S signal paths that are used between the UNIBUS and a subscriber work station are shown in more detail in Figure 7. The Headend Interface Controller generates the serial data stream which modulates (to 90%) the 16 MHz pilot frequency. The 16 MHz is divided down by 48 to provide 3 microsecond clock pulses for timing out data transmissions. The use of only 90% modulation allows each receiving modem to achieve pilot frequency recovery in a phase-locked loop working at 16 MHz. Sixteen megahertz tuned cable drivers and receivers are used to put signals on and take them off the cable. The signal from the work station's tuned cable receiver is processed for pilot recovery with the aforementioned phase locked loop technique. This pilot times out both the received data and the outgoing data. In addition, it provides the 16 MHz pilot for modulating the data going upstream. On the upstream cable 100% modulation as well as shut-off at end of transmission is used to avoid the additive effect of many upstream modulators working at the same frequency. The headend need only receive and demodulate the upstream signal. The non-16MHz signal path is passed through a notched filter which blocks the 16 MHz components, and passes all others around the headend. Thus, the headend connection is transparent to all but the S&S signals.

3.3 System Timing

The basic timing used in the current experimental laboratory signaling and supervision subsystem is illustrated in Figure 8. The bit time for data transmission is a three microsecond CLOCK, which is derived by scaling the 16 MHz pilot frequency by a factor of 48. This 333.333 kHz signal becomes the "times sixteen clock rate" used

IA — 44,557

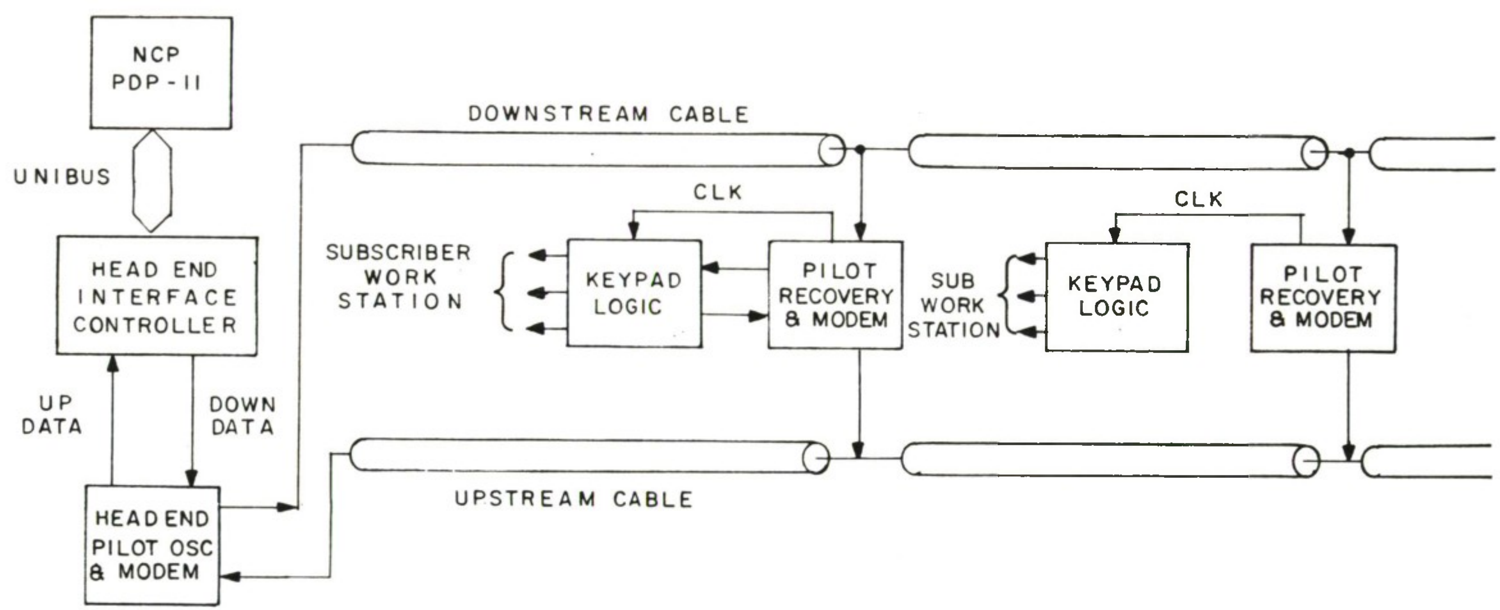


Figure 6 MAJOR SIGNALING AND SUPERVISION SUBSYSTEM COMPONENTS

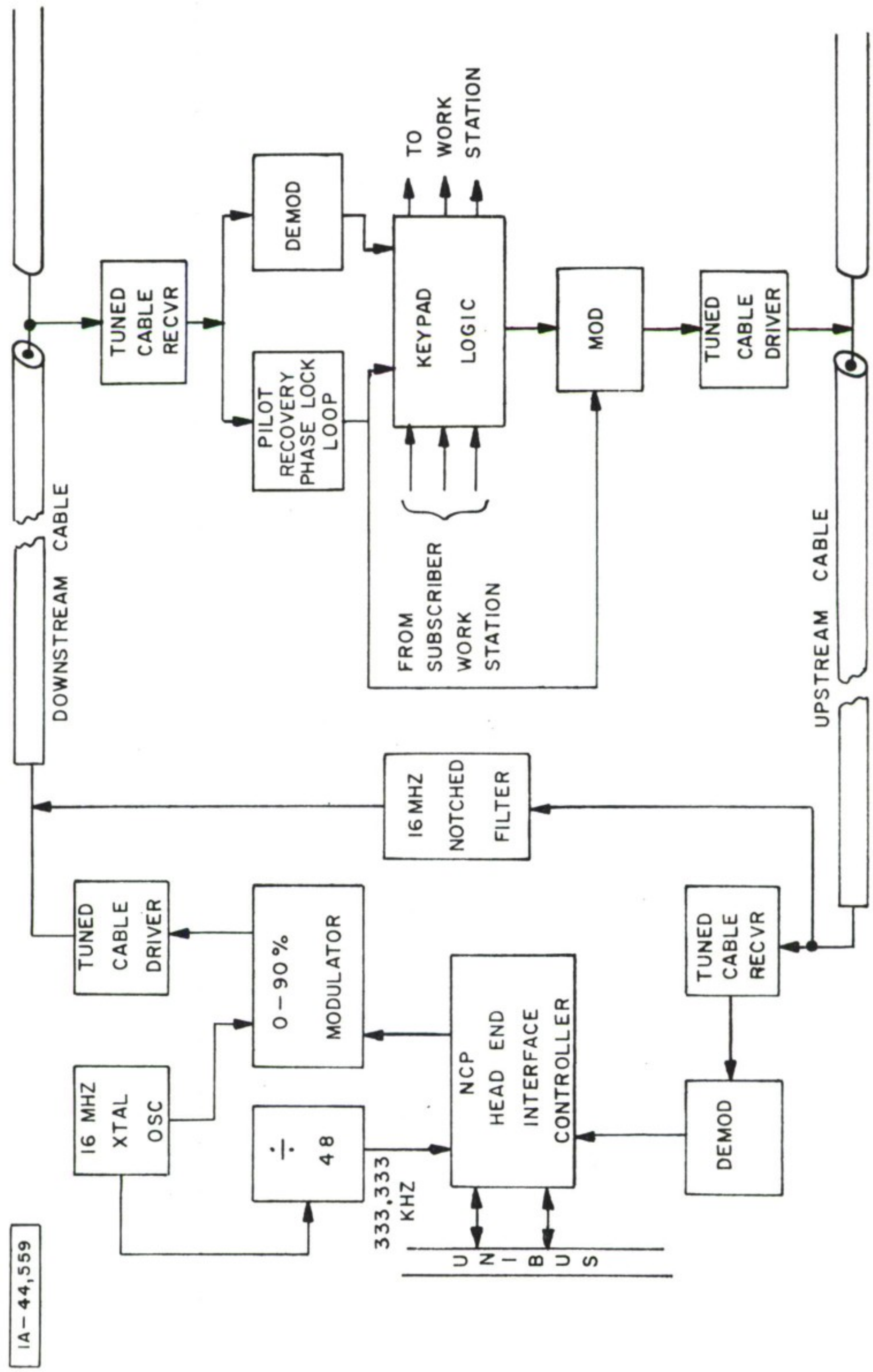


Figure 7 SIGNALING AND SUPERVISION SIGNAL PATHS

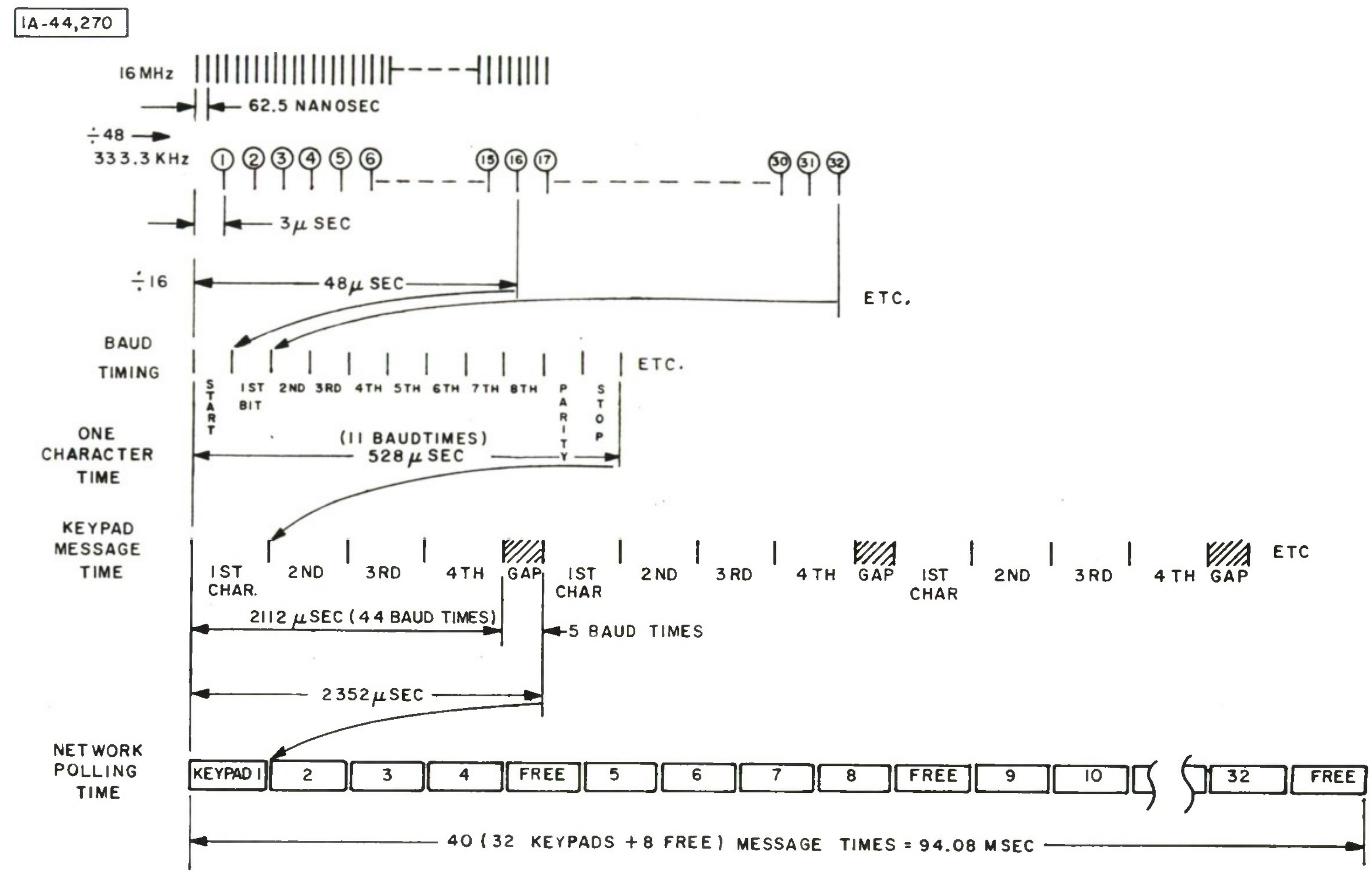


Figure 8 POLLED ASYNCHRONOUS TDM TIMING EXPERIMENTAL / DEMO SYSTEM

in the UART (Universal Asynchronous Receiver Transmitter). This creates a baud or bit time of 48 microseconds, which is approximately a 21 kilobit/second rate. The basic S&S character format, which is transmitted serially, consists of eleven baud. Eight are information bits, one is an odd parity check bit, and the other two are the start and stop bits. This results in a character transmission time which is 528 microseconds long. A full S&S message is composed of four characters, whose format is described in Section 3.4, and an intermessage guardband or gap of 5 baud times. This results in a message transmission time of 2,112 microseconds to which is added the 5 baud time delay of 240 microseconds, resulting in a total effective message period of 2.35 milliseconds. The time duration of 2.35 milliseconds per message permits polling approximately 40 times in a 10th of a second.

3.4 Polling Formats

Each keypad is polled repetitively at 10 polls per second with a hardware poll. The hardware poll is used by the S&S headend controller to interrogate the keypad for its keystroke and status information. This is a cyclic, repetitive, and consecutive interrogating procedure using a basic coded message. Occasionally, the cyclic hardware polling is interspersed with control messages called a software poll directed to some particular keypad element. The frequency of use, recipient address, and message content of a software poll are entirely under the control of the NCP.

Each "software poll" message going downstream from the NCP consists of two consecutive 16-bit words as illustrated in Figure 9.

The format for Word I and Word II is as follows:

- Word I

  bits 0 thru 13 are the Address Field,

  bit 14 is unassigned and is available as a control bit for Word I,

  bit 15 is unassigned and is available as a control bit for Word II,

- Word II

  bits 0 thru 13 are "MESSAGE",

  bits 14 & 15 denote the MODE, as follows:

  "00" is hardware poll,

  "11" is software poll.

The format of the software poll (Word II) message is illustrated in Figure 10. The two examples shown are as follows: (A) lighting the lamps at the subscriber's keypad and (B) directing the connectivity closures of a switching matrix.

The two words of the software poll are transmitted during a single 32-bit time frame, as described in detail in Section 3.3 System Timing. This 32-bit time frame governs the periodicity of the polling scheme.

The hardware poll uses Word I for sending the keypad polling address (see Figure 9) but it does not transmit a message in Word II. However, to maintain the 32-bit time frame periodicity, the hardware poll transmits blanks (binary zeros) in the Word 2 position.

IA-45,249

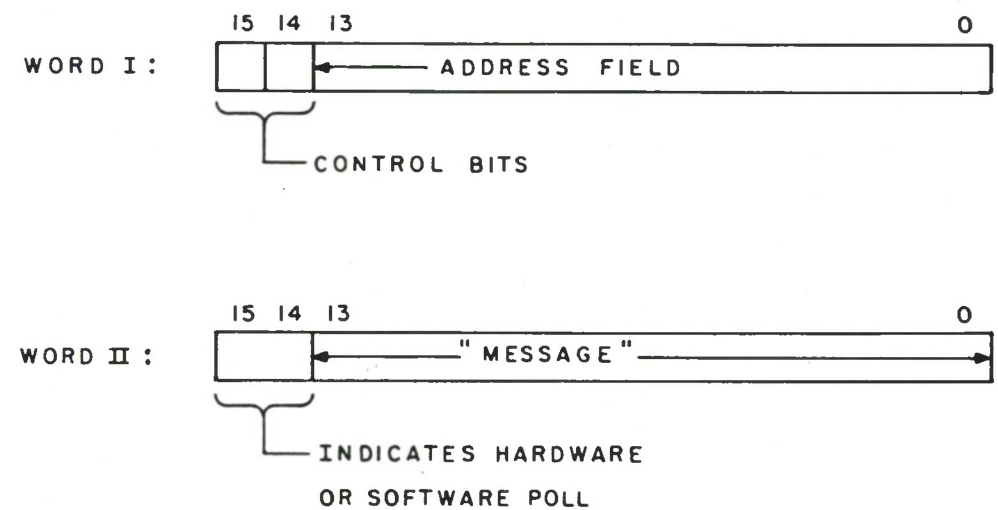


Figure 9 DOWNSTREAM MESSAGE FORMAT

IA-44,558

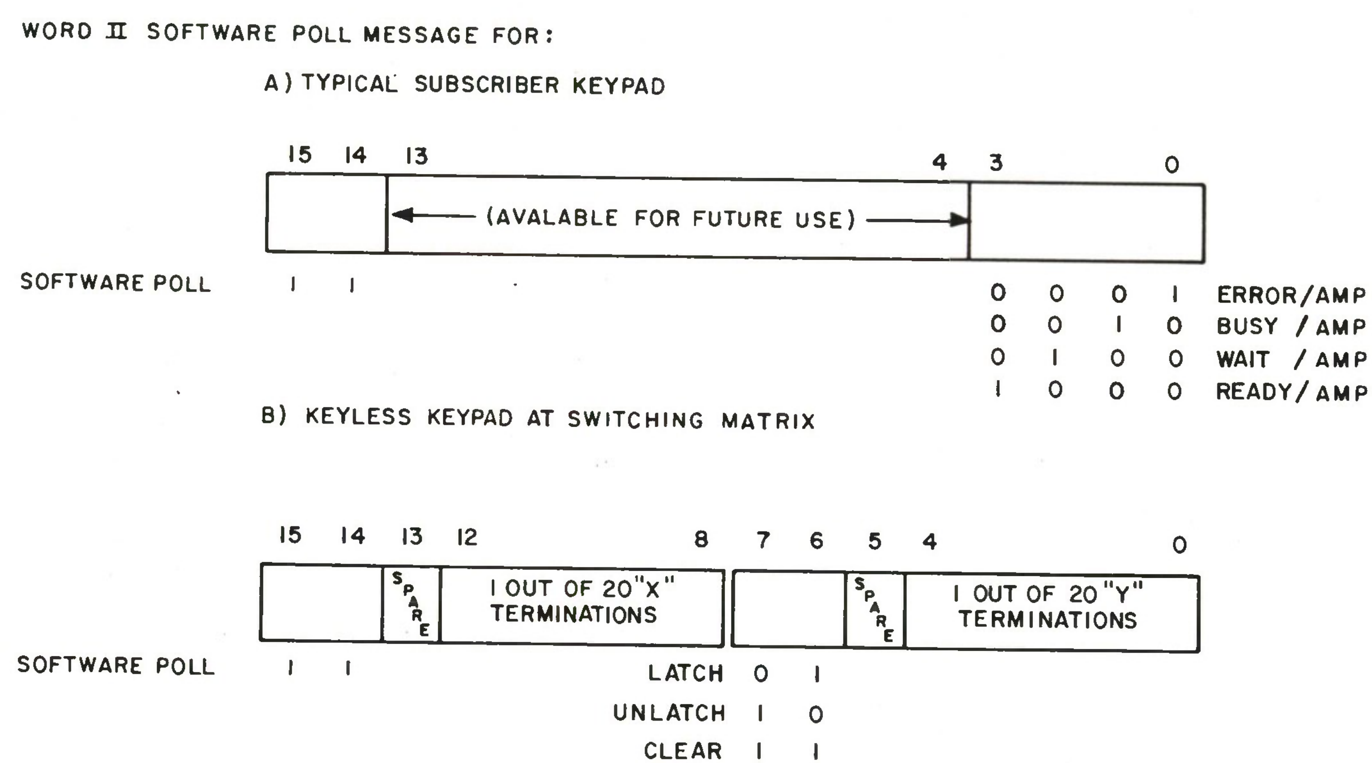


Figure 10 TYPICAL SOFTWARE POLL MESSAGE FORMATS TO A KEYPAD

3.5 Hardware Poll Responses

The response by a keypad to a hardware poll is a configuration of 24 bits, preceded by a guard band of up to 8 bit times. The format of a status response, shown in Figure 11, is as follows:

3.5.1 Guard Band

There is a guard band of time to allow for the particular cable delay that occurs in transmissions from keypads at different points along the cable. The actual delay-setting procedure takes place by a hardware vernier adjustment in the Keypad. The important point here is that the time duration that would have been occupied by the first 8 bits is dedicated to providing the guard band.

3.5.2 Check Address

The second character is an 8-bit "check address" which is the low order 8-bit set of the possible 14-bit address field. It functions like an echo check taken only over the low order 8-bits.

3.5.3 Status Bits - Message "A"

The third upstream character is an 8-bit field whose contents are an encoded image of the status of the work station elements. It could show that a station is busy, or out of order, or functioning in a certain mode. It also is used to indicate that an error was detected in the downstream transmission to that keypad workstation. The precise assignment of bits is made at the time when the type of work station it must support is defined.

3.5.4 Status Bits - Message "B"

In the standard KEYED keypad this character field is encoded with three facts. One is the last keystroke hit on the 16-button keypad. The second is a 2-bit set that says that there has been a change in the keystroke being read or a change in work station status (see paragraph 3.5.3). The third is a 2-bit field to show status weight that may be used for future encoding of additional status information.

3.6 The Keypad

The implementation of the S&S system makes use of keypads of two basic types: an active or KEYED keypad, and a passive or KEYLESS keypad.

IA-44,560

THE RESPONSE FORMAT IS :

8 BIT-TIME GUARD BAND	8 BIT-CHECK ADDRESS	8 BIT-STATUS MESSAGE "A"	8 BIT-STATUS MESSAGE "B"

FOR TYPICAL SUBSCRIBER STATION :

"A" USUALLY REPORTS STATUS OF STATION ELEMENTS

"B" REPORTS AS FOLLOWS :

4 BITS - MOST RECENT KEYSTROKE

2 BITS - CHANGE INDICATORS:

KEYSTROKE: ONE BIT

WORK STATION STATUS: ONE BIT

2 BITS - STATUS WEIGHT

Figure 11 STATUS REPORT FORMAT FROM A KEYPAD

The KEYED keypad contains a 16 button keyboard and allows a subscriber, at a work station, to key in a request for service. It also has a group of indicators to advise the subscriber about the condition of his request for connectivity. These indicators include:

- READY, denoting that the system is ready to accept S&S signals.
- WAIT, denoting that the requested connectivity is being made.
- BUSY, denoting that the requested connectivity is already in use.
- ERROR, denoting that an invalid connection was requested.

The configuration of the experimental keypads is shown in Figure 12. The 16 buttons, 4 lights and cable connections on the rear of the unit are clearly visible. An additional function of the keypad is to permit control of communication between the Network Control Processor and the subscriber work station. The NCP can address "messages" to a work station via the keypad, and the NCP can receive status reports from the work station.

The KEYLESS keypad is used for system elements where normally no operators are required. Examples of system elements that would use a KEYLESS keypad include video switches, microfiche equipment, framegrabbers, etc. Typically, these elements are addressed with control "messages" and can report their status.

3.7 Keypad Logic Design

3.7.1 Keypad Downstream Logic Structure

The Keypad's downstream logic structure is shown in Figure 13. The DATA and CLOCK signals go from the demodulator to the UART. A serial parallel character conversion takes place as part of the data handling done by the UART. The message is synchronized by use of the 5 baud time gap, and the 4 characters are timed out in conjunction with a four stage Downstream State Timer (DST). During DST 1 the high order part of the downstream address is examined. In state 2, the low order part of the downstream address is examined. If a valid address is not received, the state counter is reset and waits for the next message synchronization signal to initiate the examination of a new address code. When the correct address is

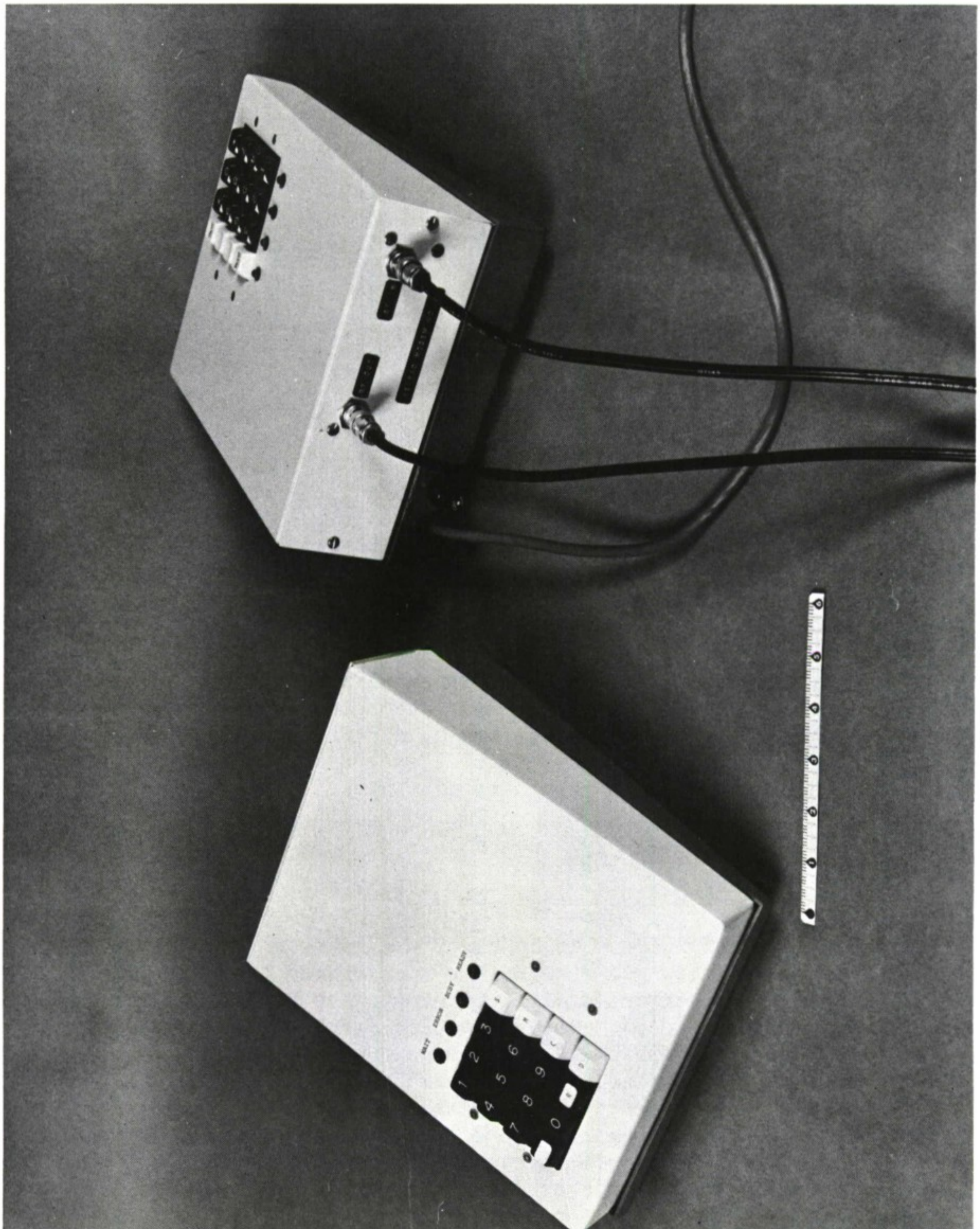


Figure 12 Signaling and Supervision Keypads

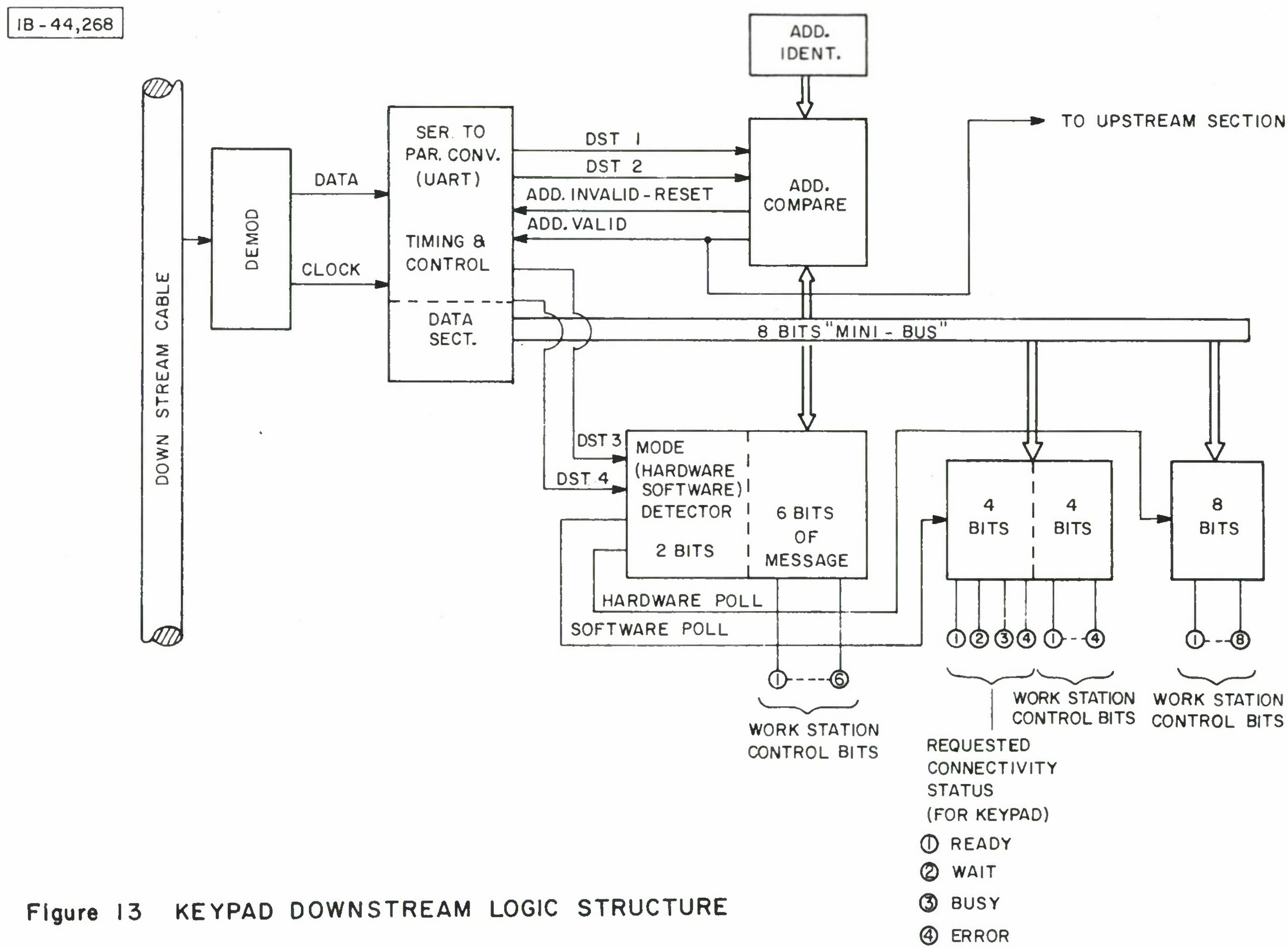


Figure 13 KEYPAD DOWNSTREAM LOGIC STRUCTURE

received, the system is put into DST 3 via the "ADDRESS VALID" signal during which time the upstream state counter is initiated and generates Upstream State Time (UST) 1, discussed below in Section 3.7.2 With DST 3, the first character of the message word was read and decoded. The latter function indicates whether a hardware or software poll is being made. DST 4 reads out the last 8-bit character. Note that any or all 14 bits (two characters less the two mode bits) can be read out in a variety of optional ways. These include, but are not limited to: control bits to a work station either under hardware or software polling or alternately, as illustrated, 4 bits which are part of the DST 4 character. These are read out during a software poll only to light one of the four status lights discussed in Section 3.6.

3.7.2 Keypad Upstream Logic Structure

The upstream logic structure, as illustrated in Figure 14, employs a bi-directional, 8 bit mini-bus. As previously noted, the ADDRESS VALID pulse from the downstream logic triggers the two functions of the upstream timing control. It implements the guard band delay for upstream transmission and steps the Upstream State Timer along to gate the character groups into the UART. There are three steps:

1. UST 1 reads out 8 check bits,

2. UST 2 reads out 8 "new" status bits and

3. UST 3 reads out 8-bit group containing status weight (2 bits), change information (2 bits), and keystroke (4 bits).

The resulting serial output is modulated onto the upstream cable. Furthermore, it will be noted that in UST 2, the present value of device status was compared with the former value of the device status and if a variation or change has been detected,then, the 3rd bit of the last character byte is set.

3.7.2.1 Status Comparison Technique. Figure 15 illustrates the organization of the bit storage and comparison procedures used to handle the 8 status bits that characterize the work station status. All actions take place in UST 2. On the initial occurrence of UST 2 any current status is loaded into a "new status" register and read out to the minibus during UST 2. (The minibus is an 8-bit bi-directional computer type circuit for centralized data flow). Via the minibus, this value is made available to an 8-bit

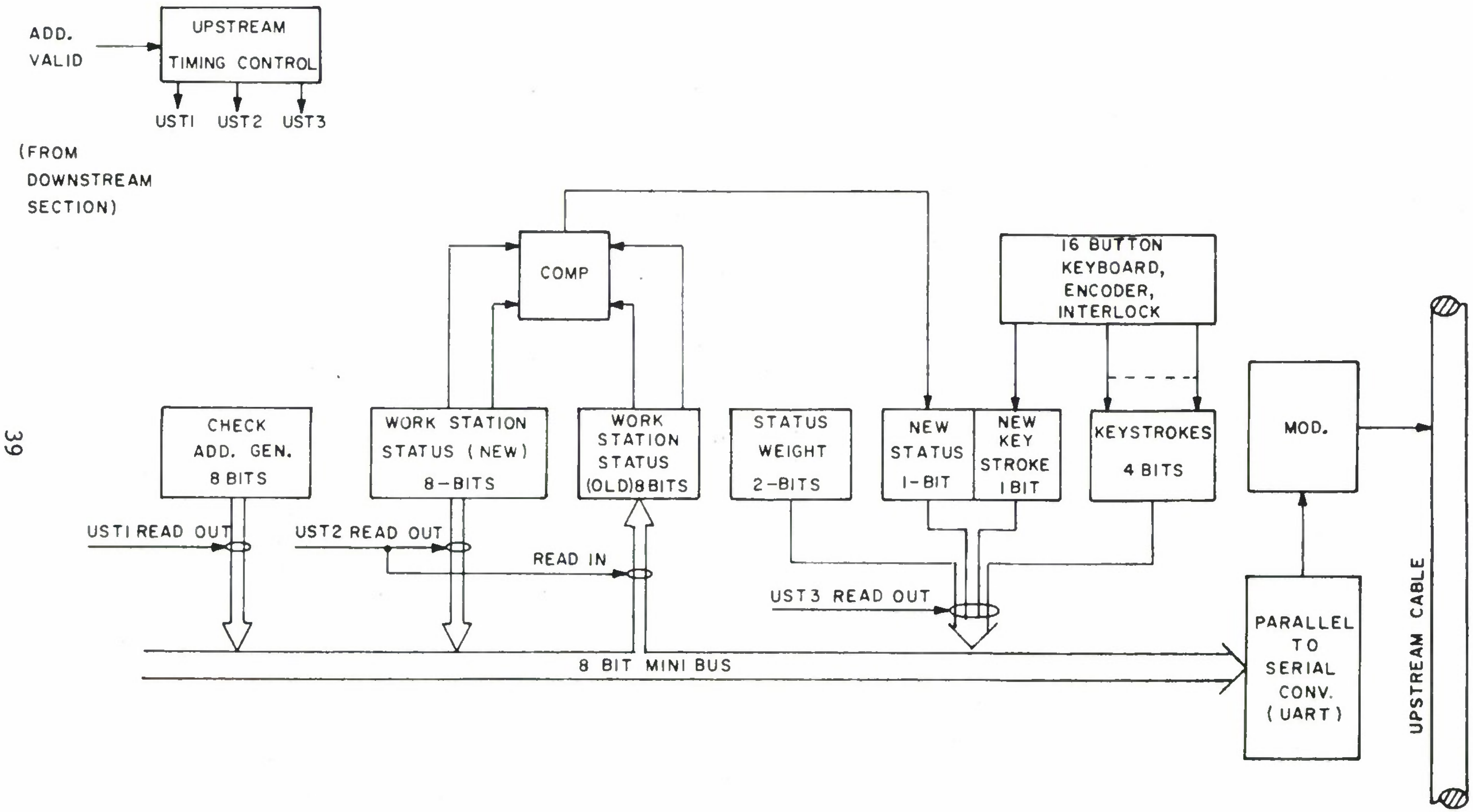


Figure 14 KEYPAD UPSTREAM LOGIC STRUCTURE

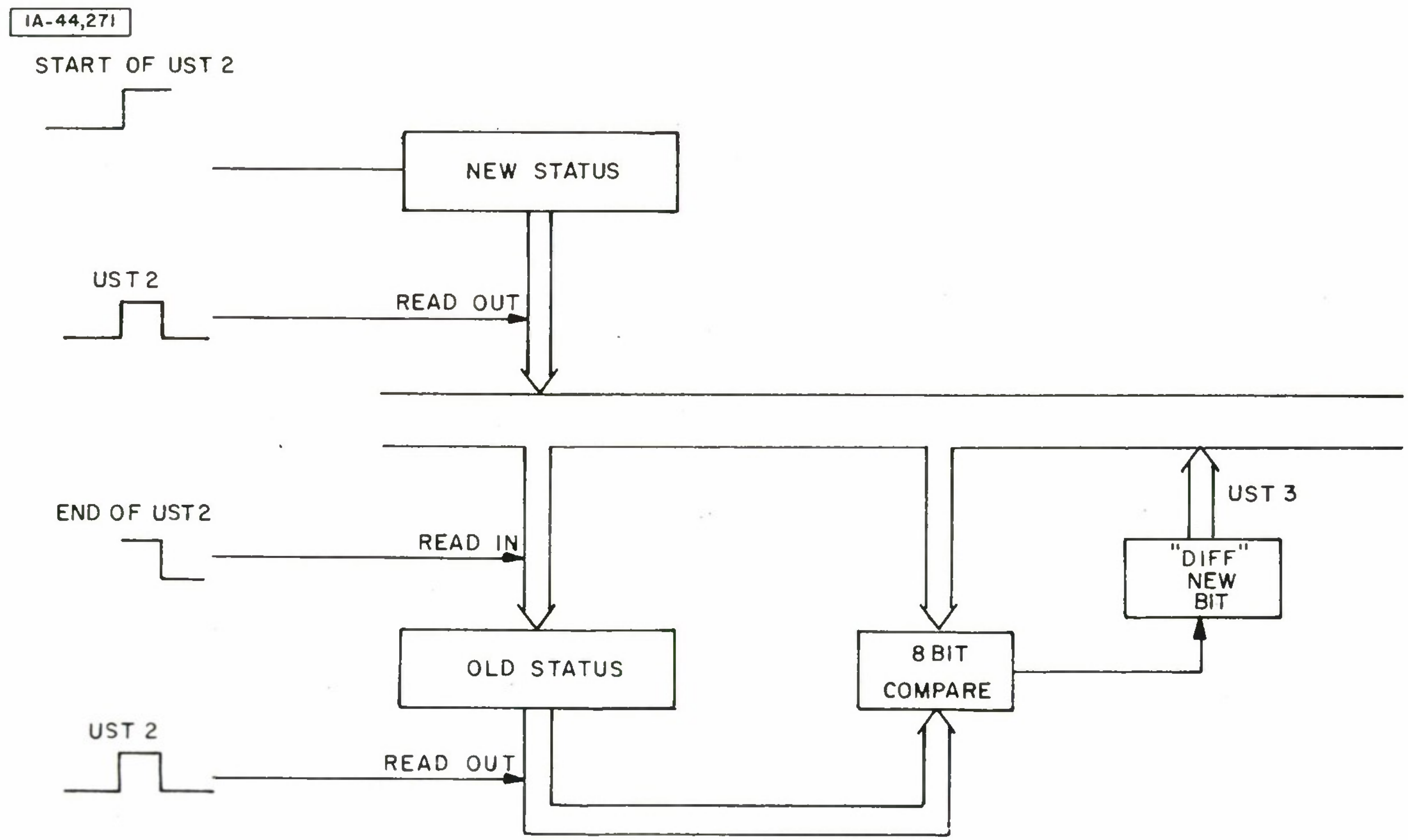


Figure 15 LOGIC STRUCTURE FOR DETECTING CHANGES IN THE STATUS BITS

comparator. The other input of the comparator is stored as the "status change" bit. On the falling edge of UST 2, the contents of the minibus are read into the "old status" register. Therefore, the "new" and "old" registers use tri-state outputs and edge-clocking resulting in a minimum parts count.

3.7.2.2 Keystroke Comparison Technique. In a similar manner, and simultaneously, a test is made to determine if the current value of the keystroke value is new since the last poll. If it is, then bit 4 is set. Note that the keystroke test is for "newness" not "difference". This technique allows the same key to be activated two or more times in succession, as required for multiple repetitions of the same character.

3.8 Headend Interface Controller

3.8.1 Headend Interface Control Hardware

The Headend Interface Controller (HIC) provides the interface between the Network Control Program (NETCOP), a software package on a PDP-11/10 minicomputer, and the keypads located at each work station. The PDP-11/10 minicomputer is shown in Figure 16.

The hardware for the controller is shown in Figure 17. It consists of two large printed circuit boards and several smaller printed circuit boards on which are mounted a mix of MSI and LSI components. The printed circuit boards plug into a DEC PDP-11 "System Unit" which is hardware designed to facilitate customized interfacing to the PDP-11 UNIBUS cable. This cable is a flat white ribbon cable clearly visible in the lower right side of Figure 17.

Functionally the HIC has one section of logic and storage dedicated to downstream polling control, and a second section dedicated to upstream message handling. The principal functions of the downstream controller, the lower board in Figure 17, include the formatting and generation of hardware polls, sequential storage of downstream software polls in a FIFO (First-In, First-Out) buffer, and formatting of interspersed software polls into the flow of downstream hardware polls. Some of these control functions are automatic, and some are under program control. Appendix A details the operator codes and various control bits used.

The upstream section appears in Figure 17 as the upper board mounted on a card extender. The principal functions of the upstream section are to receive and reformat upstream messages into 16 bit words, to note if the status change or new keystroke bits are set,

Figure 16 PDP-11 Network Control Processor and Rack

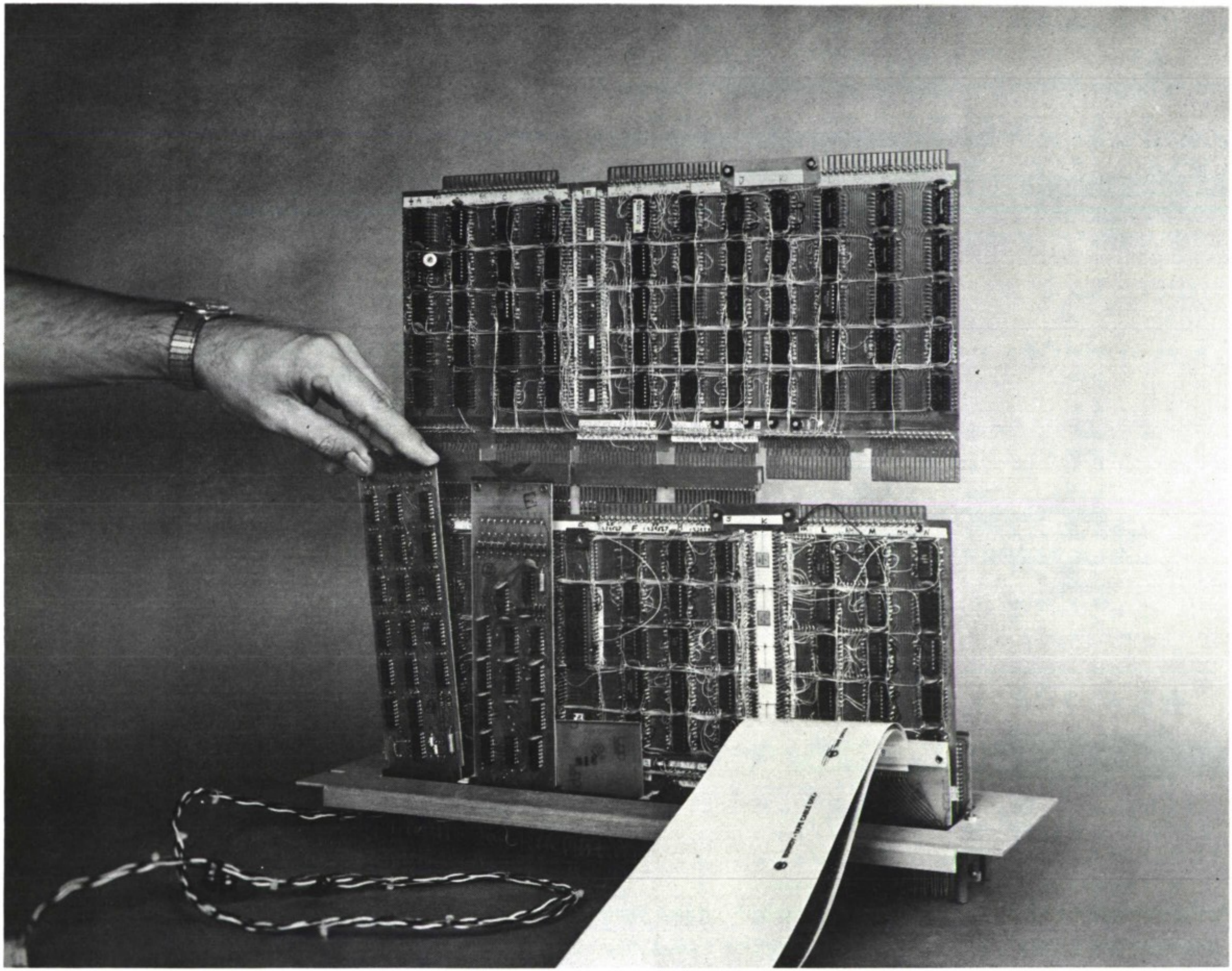


Figure 17 Headend Interface Control Hardware

and to queue such reports in a FIFO. The presence of at least one upstream message stored in the FIFO is signalled to the NCP by the HIC setting a Ready bit in its status and control register. The NCP detects this change, via the UNIBUS interrupt structure, and reads the message from the FIFO into its core memory, for interpretation and processing by the resident network control program (NETCOP).

3.8.2 Downstream Interface

Figure 18 is a block diagram of the PDP-11 headend downstream interface logic. It consists of a status and control register into which controller codes are written. The control register can also be monitored for indications of functions that relate to the proper operation of the downstream logic. Figure 18 shows data arriving via the UNIBUS and the Input Buffer that sorts out Word I and Word II for storage into a set of First-In/First-Out (FIFO) buffers that can be gated in the UART. The UART and its related timing control advance a downstream state counter that generates the timing controls DST 1 thru 5. The UART and the DST operate continuously, and on every DST 5, the Hardware Poll Address Counter (HPAC) is advanced. The low order 8-bits are read out during DST 1. Note that the high order part is not used in this system, but the controls and time slot (i.e. the architecture) exist for it. The HPAC periodically advances the Software Poll Control (SPC) conditional on the value in the FIFO Level Monitor (FLM). The FLM is incremented each time a software poll is entered via the UNIBUS and decremented every time a software poll is emitted via the UART onto the downstream cable. Thus the FLM keeps track of how many words have been read into the FIFO and how many words have been read out of it. When the FLM is at zero, no software polls are performed. Otherwise, it periodically switches the DST functions to gate on the next FIFO stored message. The FLM also indicates to the Status and Control Register (SCR) if the FIFO is 1/4, 1/2, 3/4 or completely full. The SCR enables an interrupt for these conditions. Polling is more clearly evidenced by examination of the timing diagram (Figure 8) that shows the relationship for a 40 keypad system where 32 hardware polls are interspersed with 8 software polls per poll cycle. Under software control, a software poll can be accomplished after every 2, 4, 8, or 16 hardware polls. An average rate of eight times per poll cycle is presumed to be the normal case. Under certain conditions, where it is necessary to emit many software polls rapidly, it is possible to software poll once every two hardware polls. In instances where there is little message traffic, the lower polling rates can be used. The definition and assignment of the status and control bits are given in Appendix A.

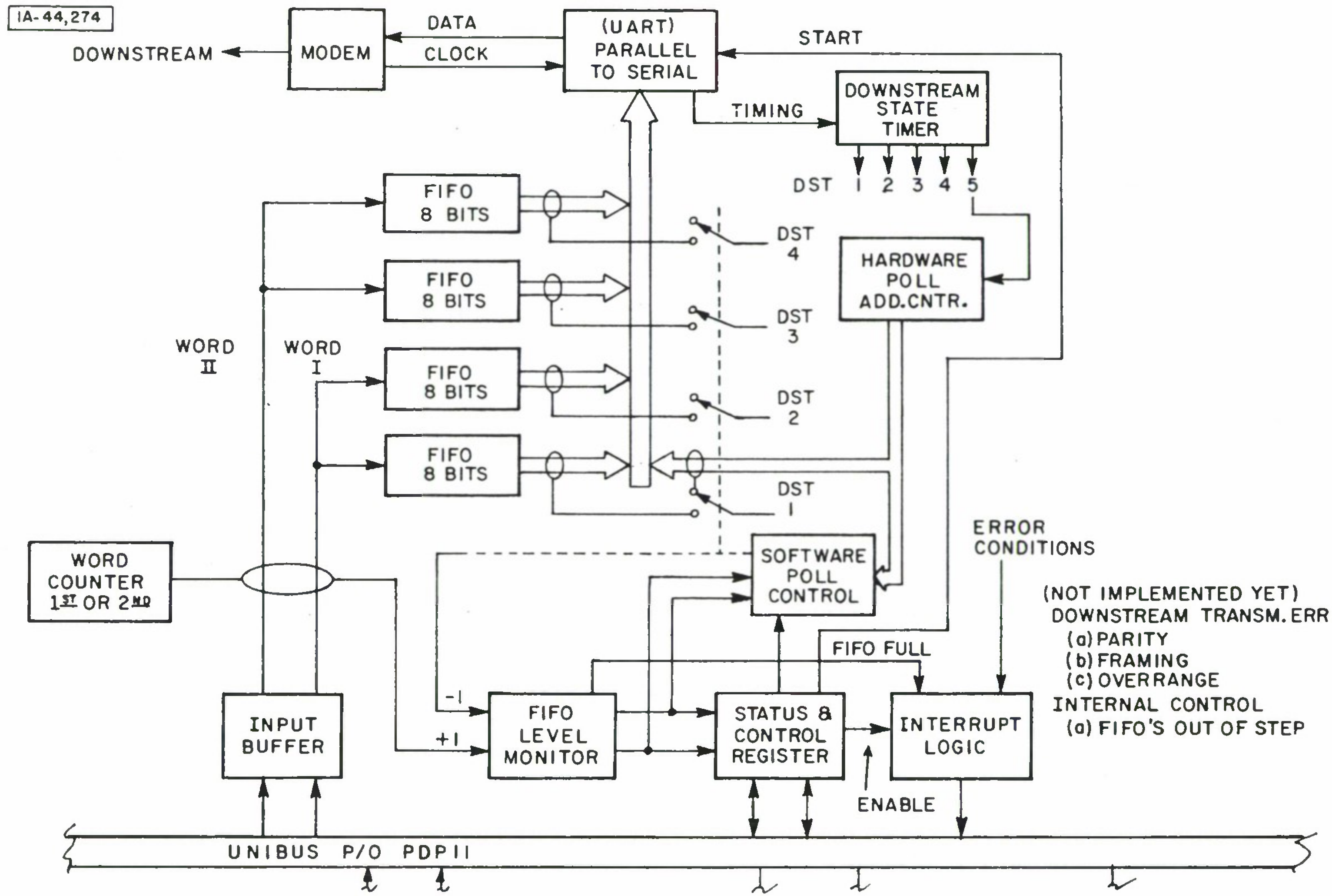


Figure 18 DOWNSTREAM SIGNALING AND SUPERVISION HEADEND

3.8.3 Upstream Interface

The upstream segment of the headend interface logic is shown in Figure 19. It is physically and electrically independent of the downstream segment, but receives 8-bit parallel signals from the UART in the downstream headend along with timing signals that provide character and message syncing. The data signals are queued up into FIFOs to provide two sequential word transmissions to the PDP-11 processor through an addressable buffer register. These two words are precisely the keypad address and the status message that includes keystroke information.

The technique used to keep track of which of the two words is to be read, is accomplished through a machine-readable bit called the Buffer Word Count (BWC). The BWC is normally set to zero, and after the first of two words is read into the PDP-11 processor, the BWC automatically changes to one. When the second word is read out, it reverts back to zero. This technique insures keeping the two word read-in process in step. The upstream logic also has a status and control word. Its format includes indicators for the FIFO buffer fullness in one quarter increments, that is, 1/4, 1/2, 3/4 or completely full. Each of these fullness levels can be used as flags to cause an interrupt when properly enabled by the program. Additionally, there is a FIFO ready bit to indicate when the FIFO contains at least one word. The ready bit also uses an interrupt enable bit. The FIFO ready bit, as well as the "fullness" bits, are developed in a FIFO level monitor for upstream use, that functions similarly to the downstream FLM. However, there are error indicators for the upstream character transmission including check address error, overrun error, framing error, and parity error. All of these are composed into one error flag bit that has an enable bit associated with it. The definition and assignment of these bits are given in Appendix A.

The key element of the logic functions of the upstream headend is the detection of bits that were preset into the status word coming upstream. These indicate a new keystroke or change in status information has been generated. These are decoded by the change bit detector. Without their presence, the word received at the headend is rejected. If either of these bits is active, then the accompanying two word message is stored in the FIFOs´ and processed as described earlier. Their presence is reported on an interrupt basis through routines that respond to the FIFO ready interrupt. A time-out circuit prevents a second FIFO ready interrupt from taking place for one millisecond. This time-out

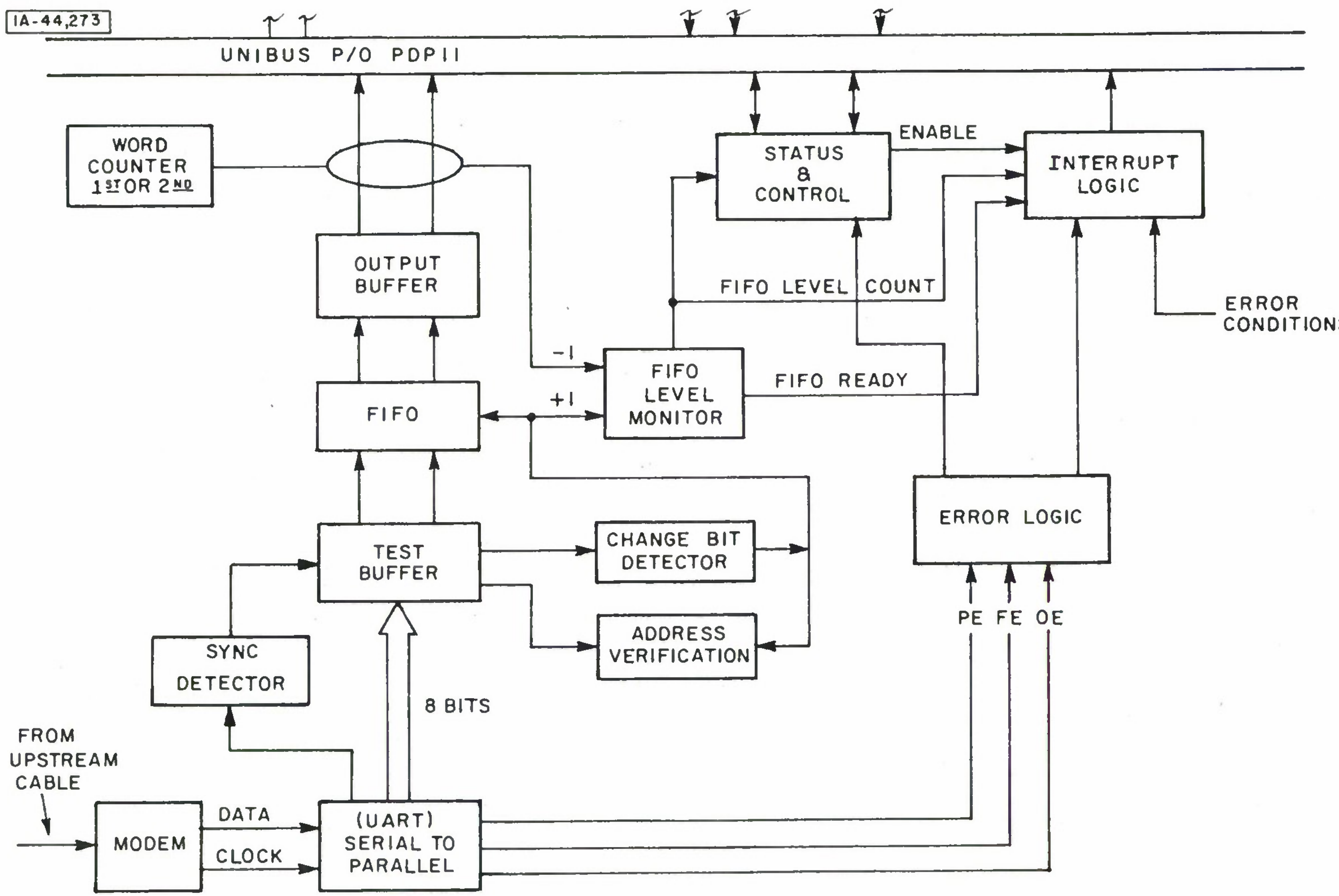


Figure 19 UPSTREAM SIGNALING AND SUPERVISION HEADEND

ensures that the NCP has at least one millisecond to process the status message before it can be interrupted again.

SECTION IV

4.0 NETWORK CENTRAL CONTROL SOFTWARE

4.1 General

4.1.1 Functional Relations

The control of the AFBITS system is provided by the software residing in the Network Control Processor (NCP). The various switching matrices and common equipments available in the AFBITS system are under the direct control of the network control processor. The processor issues the necessary commands to effect the desired connections and will receive verification information back from these control units before it returns an indication to the terminal that the user can start transmitting. The functional relationship between the processor and the rest of the control equipment that comprise the AFBITS system is illustrated in Figure 20.

Each terminal in the system contains a keypad, status indicators, and a number of user devices such as a keyboard, a visual monitor, printer, etc. The network control processor accepts requests for service from the user keypads as well as status indications from each of the user's terminal devices. If, for example, a printer runs out of paper an automatic status indication is sent to the network control processor, and the processor then acts on these changes in status by sending the appropriate network status information back to the terminal illuminating one of the status indicators.

The processor also sends the necessary commands to the switch control units and/or the common equipment control units because of either a user's request or a device's status change. The interconnecting lines between the network control processor and the controlled equipments, as illustrated in Figure 20, indicate these functional interconnections and are not separate physical paths.

Although the AFBITS system and its network control processor are intended primarily for local base distribution, the use of a programmable mini-computer for the network control processor provides the flexibility for allowing AFBITS to interface with processors in external systems such as AUTODIN and/or the Automated Telecommunications Program (ATP) network. The terminals, switch control units and common equipments, and the resulting user service packages are described in more detail in the following sections.

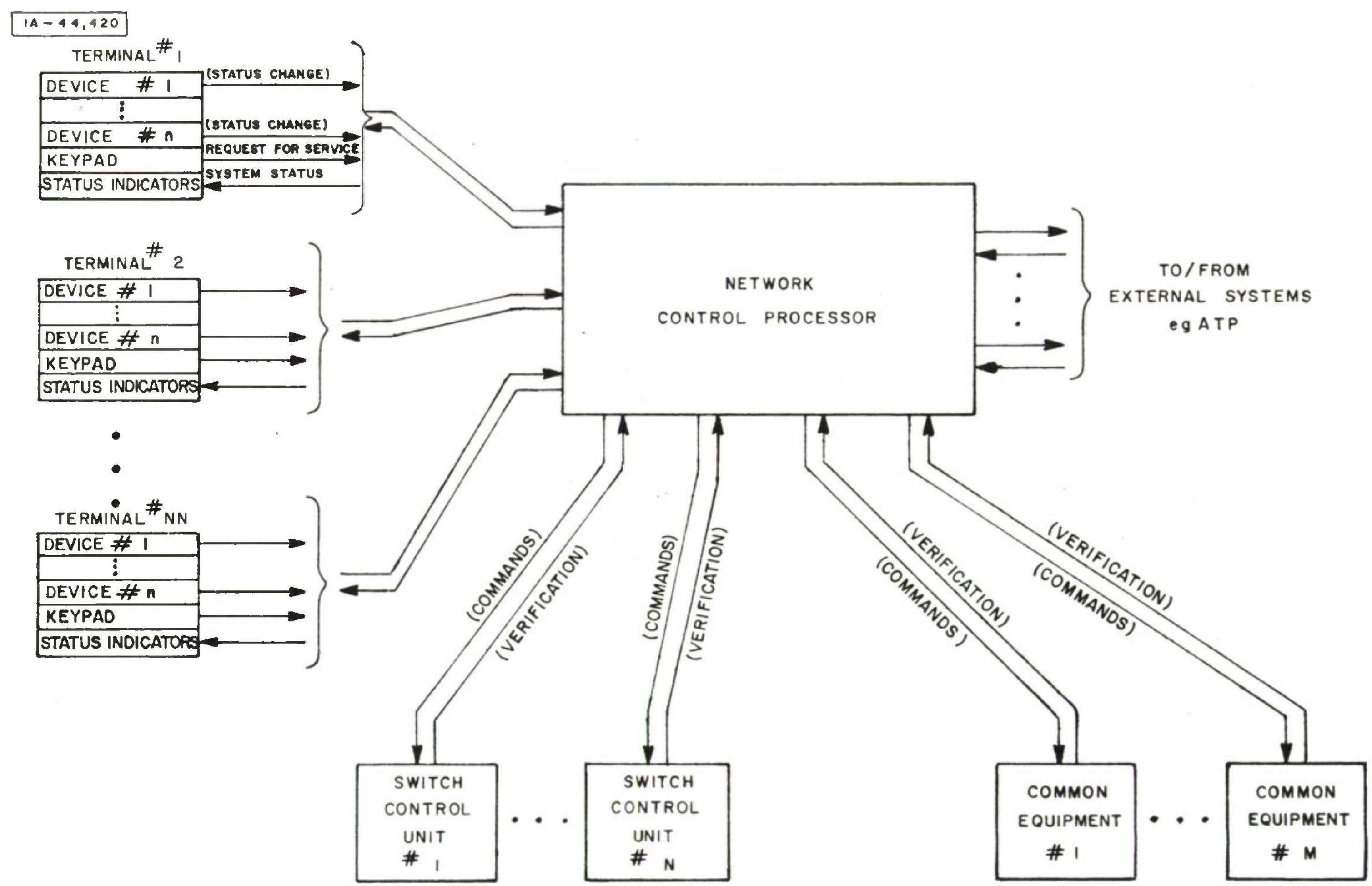


Figure 20 FUNCTIONAL RELATIONS OF THE AFBITS CONTROL EQUIPMENTS

4.1.2 Numbering Scheme

The primary input to the Network Control Program (NETCOP) are digits from the users´ keypads. On each keypad there are fifteen operational keys: nine for numerals zero through nine, and one each for "S", service package request; "M", call modify; "R", redial; "C", clear; and, "D" for disconnect.

There are five allowable dialing sequences which cause the keypad evaluation program to make connections and disconnections as follows:

1)	S XX XXXX	(service package request)
2)	XXXX	(telephone call)
3)	M X S XX XXXX	(service package modification)
4)	R	(Redial)
5)	D	(Disconnect)

These sequences are described below, where "X"s represent the ten numerals. The clear key is used any time before the end of a dialing sequence to nullify all previously dialed digits but has no effect on previously completed connections between equipments.

4.1.2.1 Service Package Request (S XX XXXX). In this dialing sequence, the "S XX" represents the service package number and the next four numerals (XXXX) designate the address. A service package request causes the processor to connect various pieces of equipment together so that the user can use the equipment at his work station for performing some function. Service package numbers and the connections made for each service package are installation dependent. NETCOP is not programmed for specific service packages, but rather has the facility for handling tables describing service packages. The tables must be set up for the installation according to the needs of the Air Force Base. For a given service package the processor can set up any number of connections between the calling party´s equipment, the called party´s equipment and common equipment. The tables within the NETCOP define these connections. The tables also contain the connections (or disconnections) to be made for legitimate call modifications.

The four-digit address portion of the service package request is either the address of another keypad or any other addressable device in the system. The tables defining the service package determine how an address is to be interpreted. A service package involving only the calling party and common equipment does not need an address, but four numeric digits must be keyed in the current

laboratory evaluation configuration or an error condition will result.

4.1.2.2 Telephone Call (XXXX). If only four numeric digits are keyed, the call is interpreted as a telephone call. Because telephone service has not yet been implemented in the current laboratory evaluation configuration, this dialing sequence now results in an error indication. When telephone service is implemented a telephone call will be treated as a service package. A service package number will be assigned to telephone service, but will not have to be keyed. Conferencing will be allowed by adding parties to the call by means of a service package modification of the form M X S XX XXXX, using the service package number assigned to telephone service.

4.1.2.3 Service Package Modification (M X S XX XXXX). Depending upon the definition of a service package in the service package tables, it may be possible to modify the service package by making additional connections. Modification will be used primarily to add additional viewers to a video connection or to create a conference telephone call. After a service package has been expanded by call modifications, the additional connections can also be dropped by call modification.

The digit following the M in the dialing sequence is the call modification number. The call modification number defines the connection to be made or broken. These connections involve equipment of the calling party, a new called party and common equipment that was used in the original service package request.

Call modification number zero is reserved for the special function of disconnecting an entire service package and all of its modifications. The dialing sequence for this is "M S XX XXXX" where the service package number (XX) is the service package to be disconnected and the address (XXXX) can be any valid digits. This operation, referred to in the flow charts as "delete service package" is used when a user wants to disconnect one service package but retain another. If a user has only one service package set up, he keys "disconnect" for the same result.

4.1.2.4 Redial (R). Whenever the processor returns a busy signal to a user, a record is retained of his keystrokes in the NETCOP queue. The dialing sequence "R" causes a re-evaluation of his original calling sequence as soon as the requested connection becomes available. The user has this redial option until he makes his connection, keys another legal dialing sequence other than

disconnect, or a system initialization intervenes. Hitting disconnect has no effect on the redial option unless the dialing sequence being held for redial was a call modification, in which case, the redial option would be voided.

4.1.2.5 Disconnect (D). The keying of disconnect causes all service packages and telephone connections set up by the user to be disconnected. If more than one service package is in operation and only one is to be disconnected, the delete service package option should be used (M O S XX XXXX). The hanging up of a telephone will be treated as a delete service package if the user was the originator of the telephone call.

4.1.3 Indicator Lights

There are four indicator lights on a keypad: Ready, Wait, Busy and Error. Only one of these will light at a time. Ready indicates the normal condition. The Ready light shows the system is operational and remains on while the user is dialing. At the completion of a dialing sequence the Wait light is lit, and will usually stay on a very short time until the result of the request is determined. After the Wait light is illuminated, either Busy, Error, or Ready indicators are lit. If "Ready," then action on the request is complete; there is no time out on the Ready light. An Error or Busy light can be extinguished by keying the CLEAR button. Otherwise it will stay on for a few seconds, then automatically change to Ready.

4.1.4 User/System Information Flow

The user/system interactions are illustrated in Figure 21 and 22. The sequential actions between the user and the system are flow-charted in Figure 21, and the hardware interactions between the user's equipment and the network control processor are block-diagrammed in Figure 22. When the user wishes to obtain a connection, he initiates the dialing sequence as described in the previous section by going off-hook or by depressing the SERVICE button or by depressing the CALL MOD button. At this time, the Ready light illuminates (if it was not already lit). The user may abort the call-attempt by going on hook, by depressing the CLEAR button, or by depressing the DISCONNECT button. The system will then erase that attempt. This feature is necessary in the event that the user gets interrupted during his dialing sequence and cannot complete the dialing sequence.

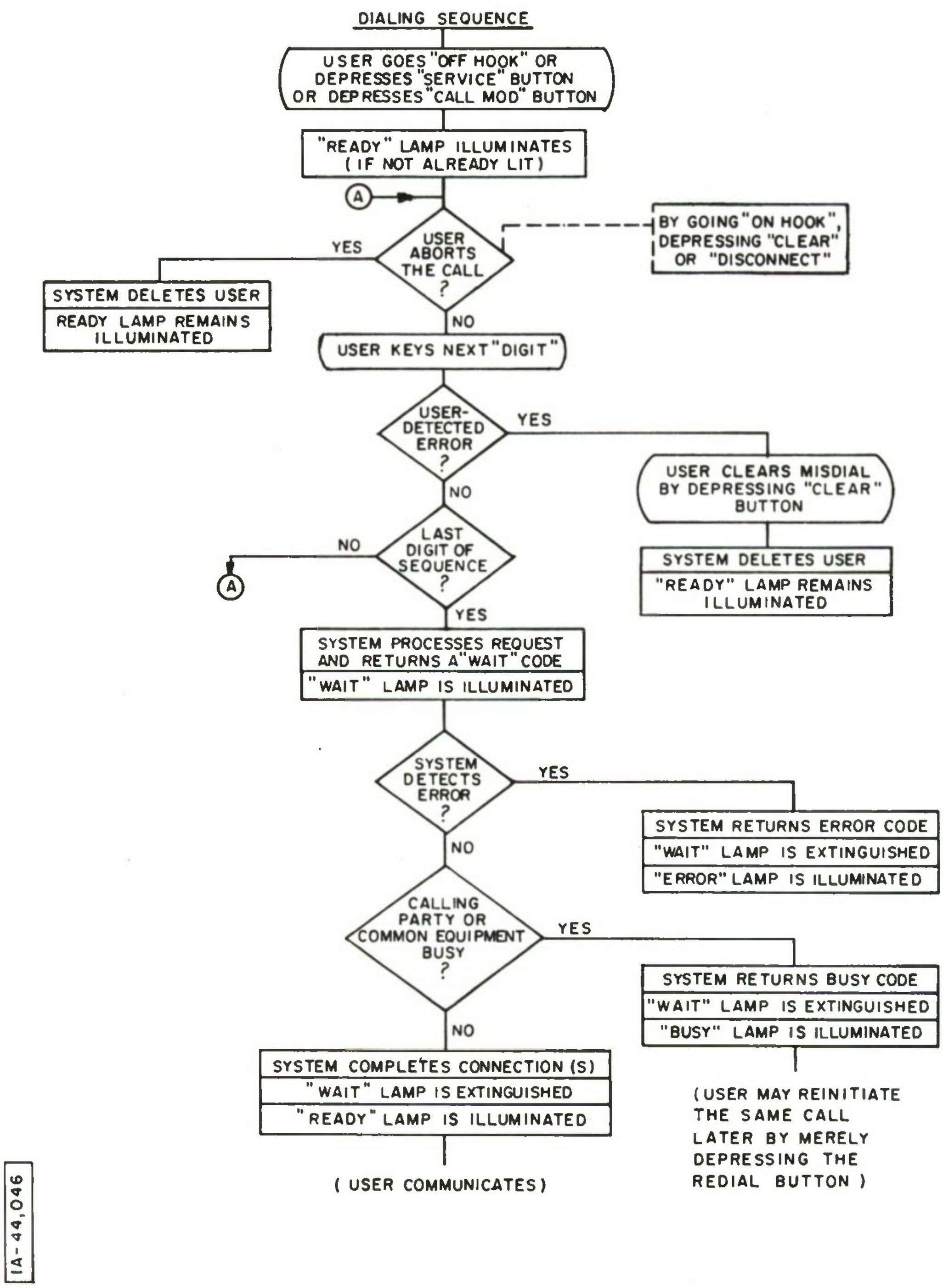
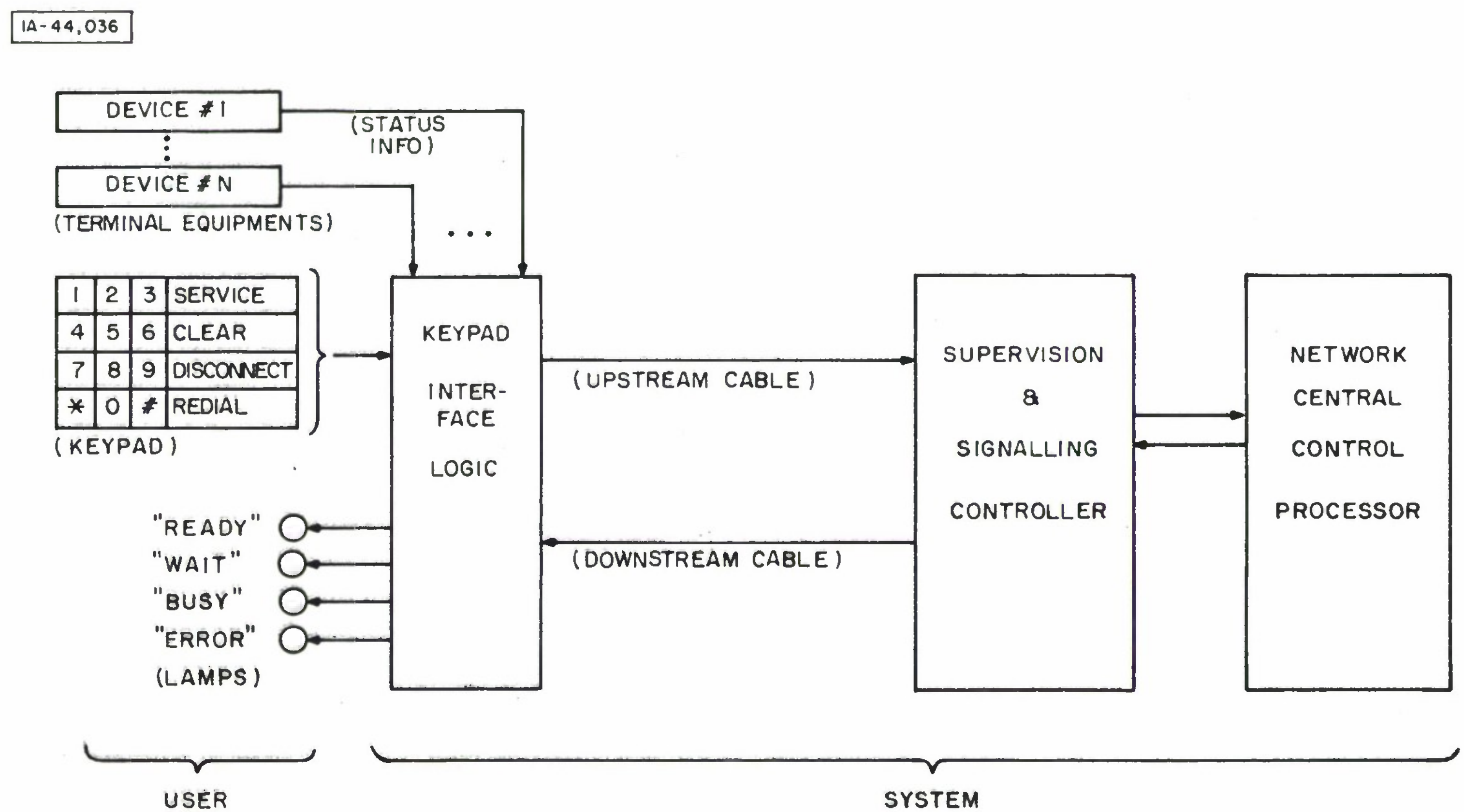


Figure 21 USER AND SYSTEM INTERACTION



* : CALL MOD BUTTON

: UNASSIGNED (SPARE) BUTTON

Figure 22 USER AND SYSTEM BLOCK DIAGRAM

Assuming he continues the dailing sequence, he keys in his next digit. If he detects that he has made a mistake during the keying in of his digits, he may clear the request by depressing the CLEAR button. When the user has completed his dialing sequence, the system recognizes the last digit transmitted and sends a WAIT signal. This does not mean that the user will physically have to wait, but since the system uses illuminated lamps and not audible tones, it merely indicates to the user that he is working into an active system.

If an error is detected in his dialing sequence when the user's request is processed, an ERROR code is sent back to his error lamp. If there is no error, the system then checks to see if either the destination (called) party or the common equipment required to complete the connection is available. If the connection cannot be completed due to a busy condition, the system sends back a busy indication to the requestor. Otherwise, the connection is completed and the WAIT lamp is extinguished. The user then communicates via his terminal. It should be noted that after each system action initiated by the requestor, the system starts a "time-out" that will result in disconnecting the user unless he proceeds to his next action within a certain period of time. In the event that the user has received a busy signal from the system, he may try this same connection again at any later time by simply depressing the REDIAL button. This is a convenience to the user who wishes to repeat his request. However, if after he gets the busy indication, he proceeds to a new dialing sequence to set up a different connection, he can no longer use the REDIAL to recall the original attempt.

Figure 22 illustrates the hardware involved in the sequence just described. It consists of a user's keypad that has keys for ten digits and six functions, for a total of 16 keys. Five of the six functions are SERVICE, CLEAR, DISCONNECT, REDIAL, and CALL MODIFICATION. The sixth function key is unassigned and is available for future use. The keypad also contains the associated status lamps that indicate READY, WAIT, BUSY, and ERROR states, as previously described. The keypad requests are transmitted upstream via interface logic located within the keypad. In addition, device status is also accommodated by the same interface logic. In effect, the complete status of the keypad and associated terminal devices is handled by the single logical interface. The logic required for driving the status lamps from the downstream channel is also contained in the same interface.

The keystroke information and device status information are both transmitted in an appropriate time slot in the upstream direction as

described in Section 3. The information contained in the time slot is examined by the Signaling and Supervision (S&S) controller located at the headend of the cable system to determine if a change has taken place since the last upstream transmission. If a change in status has occurred, the S&S controller sends an "interrupt" to the network control processor via the minicomputer interface. The processor then services the "interrupt" and performs the necessary software functions as further described in this section. Ultimately, this will result in sending signals back to the keypad interface to change the state of one of the lamps. This is done via the minicomputer interface to the S&S controller, which inserts the message into the next downstream time slot.

4.2 Program Design

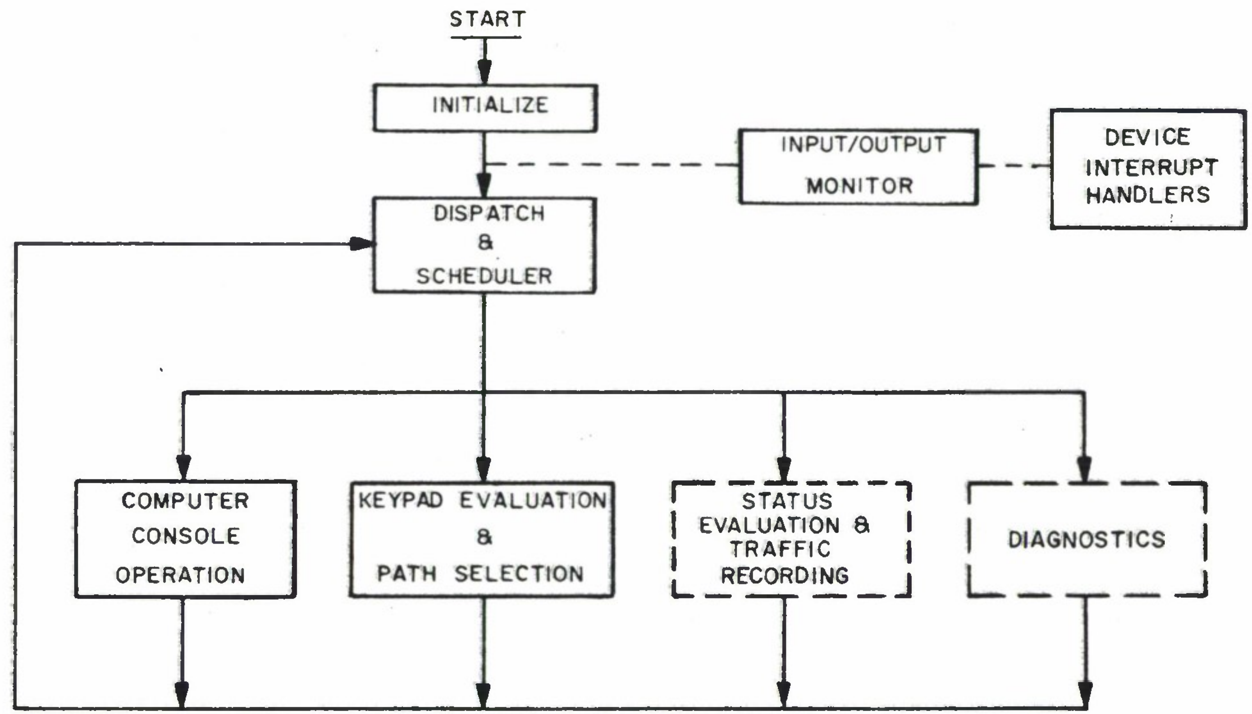
The Network Control Program (NETCOP) for the AFBITS laboratory demonstration model is a real-time, assembly language program written for the PDP-11 minicomputer. Its purpose is to accept requests from users, interpret these requests, and send appropriate commands to matrix switches, common equipments and user terminals.

The overall program structure is illustrated in Figure 23. To allow expansion and change without rewriting the entire program, the program logic was organized into modules. Each module has a well-defined function and interface with all other modules. As indicated in the figure, program control is passed to Dispatch upon completion of system initialization. Dispatch, in conjunction with the Scheduler, selects one of the following modules for execution: Keypad Evaluator and Path Selection, Computer Console Operation, Status Evaluation and Traffic Recording, or Diagnostics. When the selected module has completed execution, program control is returned to the Dispatch module.

Independent of the operating program modules are the Device Interrupt Handler modules. Once the system is initialized any peripheral device may attempt to interrupt the processor, regardless of which program is in execution, via the processor's hardware - interrupt structure. If the peripheral's priority is higher than the running program, the processor will pass program control from the running program to the peripheral device program. Upon completion of the peripheral device's program, control will pass back to the operating program previously in execution. Since sufficient data was automatically stored by the processor in the memory "stack" at the time of interrupt, no disruption in operating program execution occurs. In Figure 23, the Device Interrupt Handler program modules and their associated supervisory Input/Output Monitor program module are shown connected to the main program by dashed lines to differentiate the interrupt-driven process from the normal program flow process.

The balance of this section is devoted to a detailed description of the software program modules; additonal details on the PDP-11 minicomputer are contained in Section 5. It should be noted that the program design is not intended to be restricted to any particular minicomputer manufacturer's product offering.

IA-42,748



EACH BLOCK REPRESENTS ONE OR MORE MAJOR ROUTINES

Figure 23 NETCOP PROGRAM STRUCTURE

4.2.1 NETCOP Program Modules

4.2.1.1 Initialize. Program execution begins with this module. As soon as the entire program has been loaded, the module halts, thus allowing a computer operator to be sure that all needed peripherals have been turned on. After the check is made, the computer operator depresses the CONTINUE button and the Initialize module sets all tables and pointers in all other modules to their dormant state. The computer operator may then make any entries needed in the program tables. Once all the information is entered the operator depresses the CONTINUE button again. The module will then enable the various peripheral devices and transfer control to the Dispatch module. The initialization sequence of operations is shown in Figure 24.

4.2.1.2 Dispatch/Scheduler. The Dispatch routine (Figure 25) provides for the orderly transfer of control between the modules comprising the main program. It accepts information from the Scheduler and determines which module should be run next and contains within it a table of modules that it runs in a fixed-priority order. All modules in the main program return to Dispatch.

The Scheduler (Figure 26) accepts requests from one module for the execution of another. Data from the module is transferred and saved and Dispatch is informed of the request. At a future time, the Scheduler may also alter the Dispatch decision paths as indicated in the flowchart by dashed lines. This permits the Dispatch/Scheduler to accommodate dynamically changing system loads, thereby eliminating the possibility of excessive queueing within the processor.

4.2.1.3 Keypad Evaluation and Path Selection. This module (Figure 27) consists of the Keypad Evaluation routine and the Path Selection Subroutine. The Keypad Evaluation routine accepts a completed dialing sequence from the S&S Input module. This data is checked for validity and availability of requested-system resources. If all is in order, the resources for the service package requested are allocated. When appropriate the Keypad Evaluation routine calls its subroutine Path, which selects a path through the switching matrices and the proper control codes are transmitted.

4.2.1.4 Computer Console Operator. This module in Figure 24 allows the computer console operator to search, examine, and/or modify any memory location in the computer while NETCOP is running. It is DEC's ODT-11X (On-line Debugging Technique for the PDP-11) with modifications that prevents operator interruption of the real

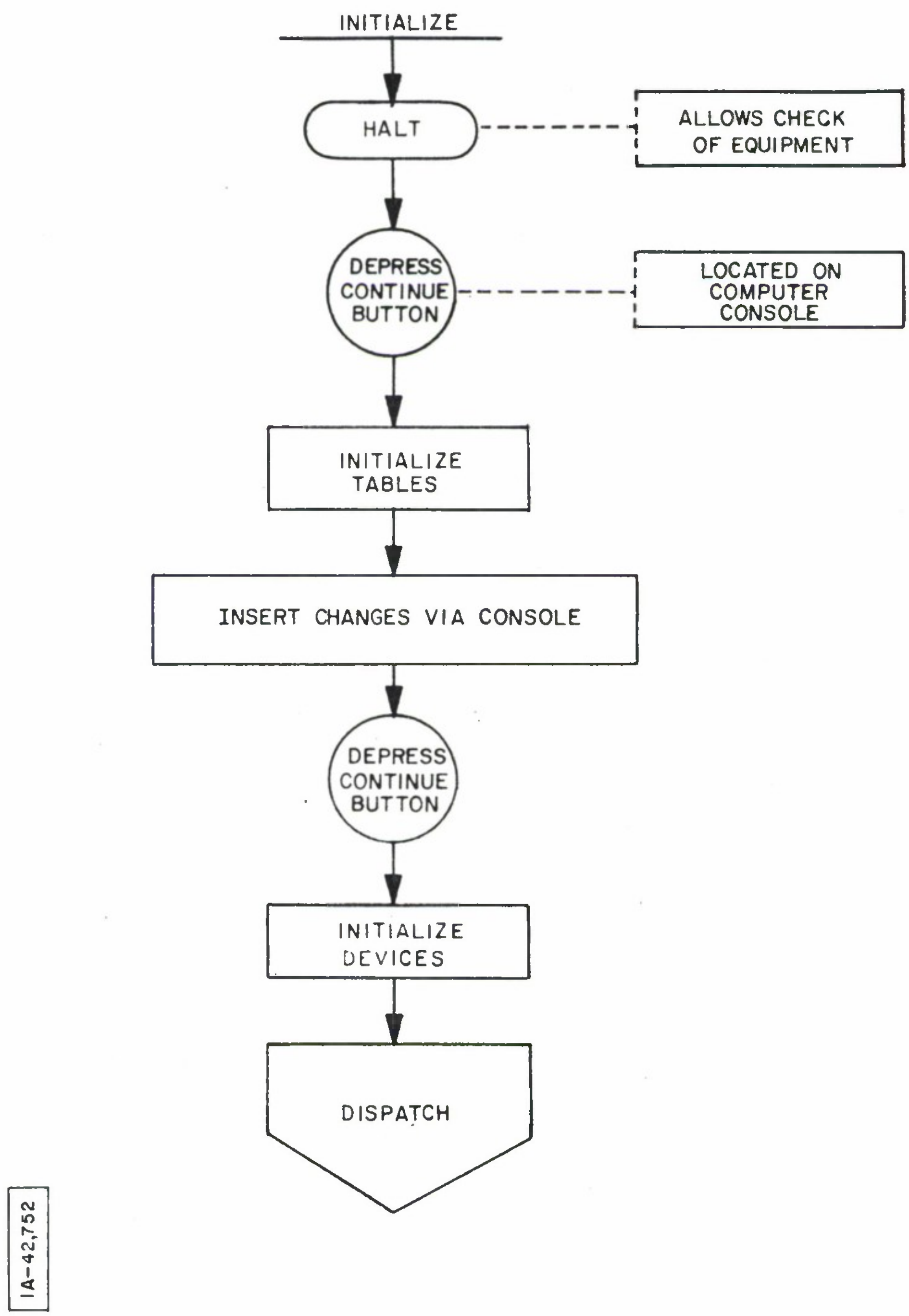
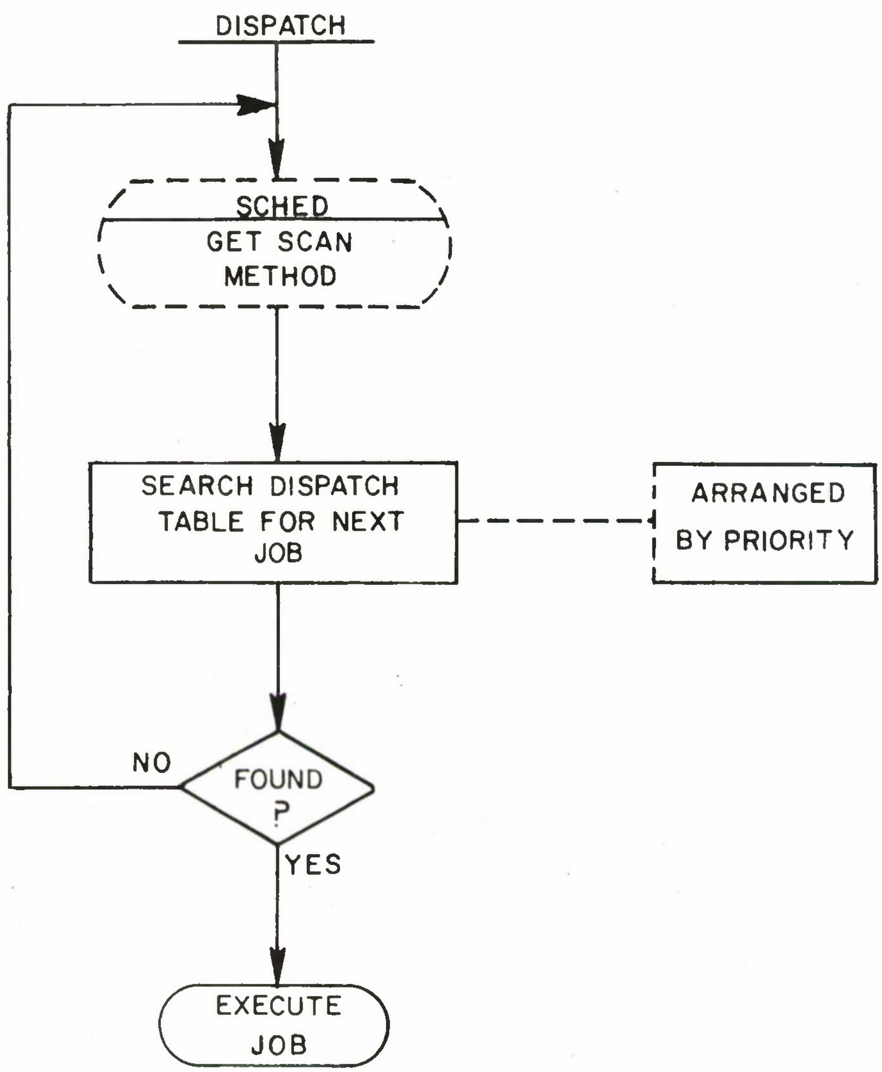
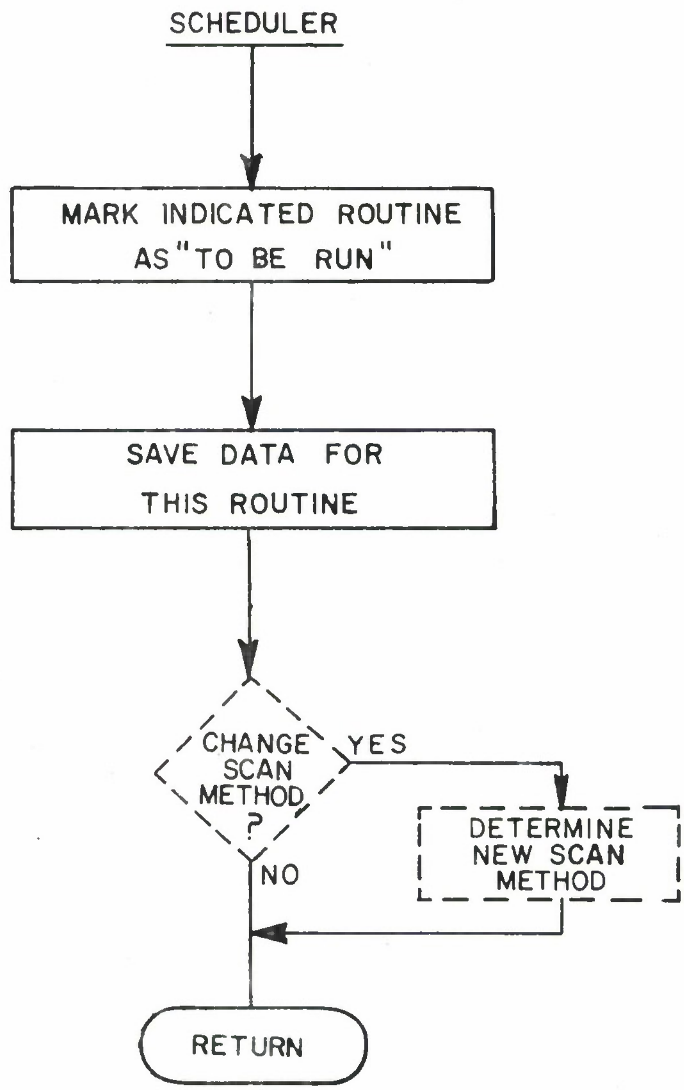


Figure 24 INITIALIZATION MODULE



1A-42,750

Figure 25 DISPATCH ROUTINE



IA-42,761

Figure 26 SCHEDULER ROUTINE

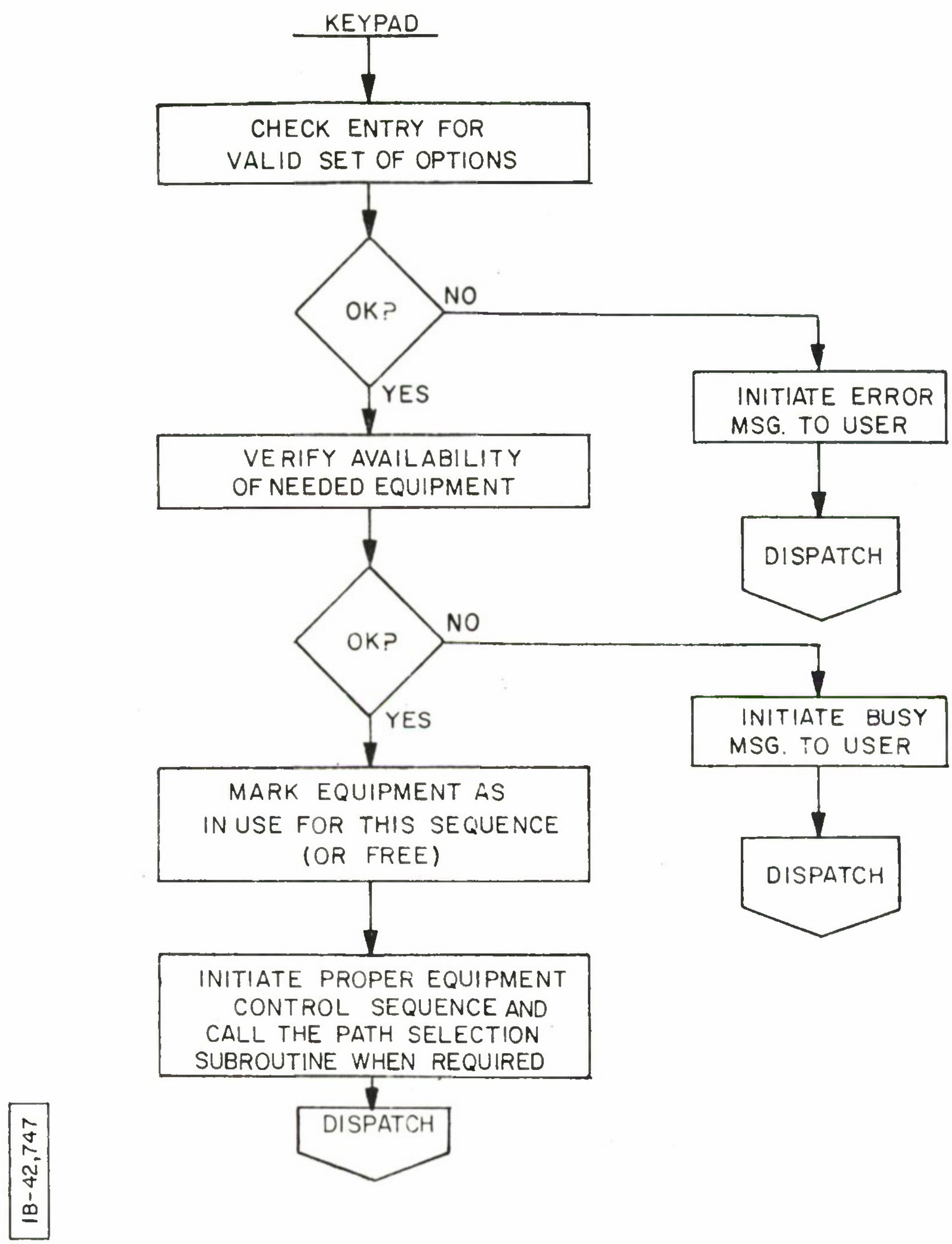


Figure 27 KEYPAD EVALUATION & PATH SELECTION MODULE

time program and uses the Input/Output Monitor module for console input/output interrupt routines. Specifically, the modifications eliminate the P (Proceed), B (Breakpoint), G (Go) and S (Disable) commands and change the program priority from seven (highest) to zero (lowest). The purpose of the module is to allow an operator to examine and change tables in NETCOP. The ODT-11X is described completely in DEC's PDP-11 Paper Tape Software Programming Handbook (Reference 2).

4.2.1.5 <u>Status Reporting & Traffic Recording</u>. This module in Figure 24 would accept status information from the various system equipments and inform the Keypad and Diagnostic modules of appropriate changes. It also records traffic metering information regarding subscriber usage of the system. Periodic printouts on status and traffic data are also included in this module. (No programming on this module has been accomplished to date.)

4.2.1.6 <u>Diagnostics</u>. The Diagnostics module would analyze hardware/software failures and keep maintenance personnel aware of the status of the system and should be capable of working interactively with them. This feature can be used as the beginning of a maintenance operation, helping to detect trouble before a system element fails completely. It may also be used to assist in verification testing of repaired suspect system elements. (No programming on this module has been performed to date).

4.2.1.7 <u>Input/Output Monitor</u>. This module is a program that monitors all input/output between the PDP-11 and the following peripherals: line printer, teletype keyboard and printer, line clock and power fail circuits. This module is a modification of a standard Digital Equipment Corporation software program called IOXLPT (Input/Output Executive with Line Printer), as described in DEC's PDP-11 Paper Tape Software Programming Handbook (Reference 2).

The modifications include the addition of the power fail routine; addition of the line clock routine; making the module relocatable, (that is changing from a fixed-starting-address in memory to a relocatable address); adding ESC (Escape) code to end formatted ASCII input; and moving character full and count bytes to correspond with CAPS 11 (Reference 3). CAPS 11 is DEC's magnetic tape cassette programming system used to assemble and load the programs for the AFBITS laboratory evaluation configuration. (The standard IOXLPT is for DEC's programming system based on high-speed paper tape input/output.)

4.2.1.8 Device Interrupt Handlers.

There is one device interrupt handler routine for each computer peripheral device. The routines for the standard peripherals are resident within the Input/Output Monitor (IOM) module. These standard peripherals include the power-fail detection circuitry, the line clock, the teletype keyboard/printer and the high-speed line printer.

The routines for handling the special peripheral, the headend interface controller described in Section 3.8, are called SASI and SASO. The SASI routine accepts the upstream signaling and supervision input messages from the headend controller. The SASO routine provides the signaling and supervison output messages to the controller for downstream message transmission. These two routines were not differentiated from the standard routines in Figure 24 for the sake of diagram simplicity. In actuality the SASI and SASO routines are not resident within the Input/Output Monitor but were written special for the NETCOP program. However, they do operate via the same interrupt-driven control logic associated with the PDP-11 UNIBUS structure.

The handlers are illustrated in Figure 28 in order of priority and are discussed below. The Power Fail circuit has the highest interrupt-level priority while the teletype printer has the lowest interrupt-level priority.

4.2.1.8.1 Power Fail. The Power Fail routine (Figure 29) consists of two parts: power-down and power-up. Upon detecting power failure, the power-down program saves all the contents of the computer registers, sets the vector location to power-up (that is it stores in a reserved memory location, called the "power fail trap vector", the address of the power up routine) and then it halts the computer. The power-up program restores this information, enables the devices in use, puts the location of the power-down routine in the power fail trap vector, and exits from the interrupt by an RTI (Return from Interrupt) instruction, thus enabling continuation of the original program.

4.2.1.8.2 Line Clock. The Line Clock routine is associated with the line-time clock, KW 11-L, which is a PDP-11 peripheral device that provides "real-time interval" interrupts. The intervals are determined by the a-c electrical line frequency, hence the designation Line Clock. For 60 Hz the intervals are 1/60th of a second, that is 16.67 milliseconds.

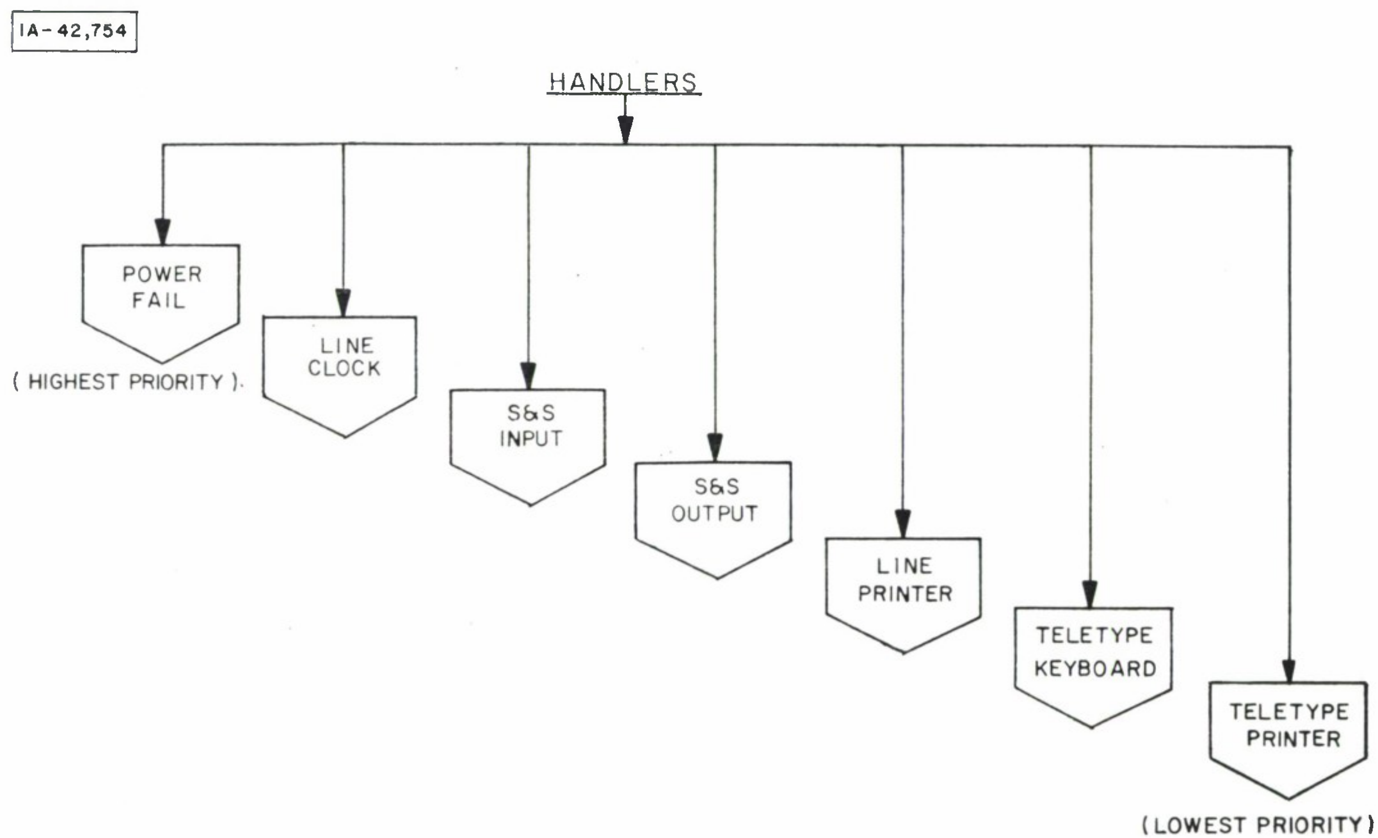


Figure 28 DEVICE INTERRUPT HANDLERS

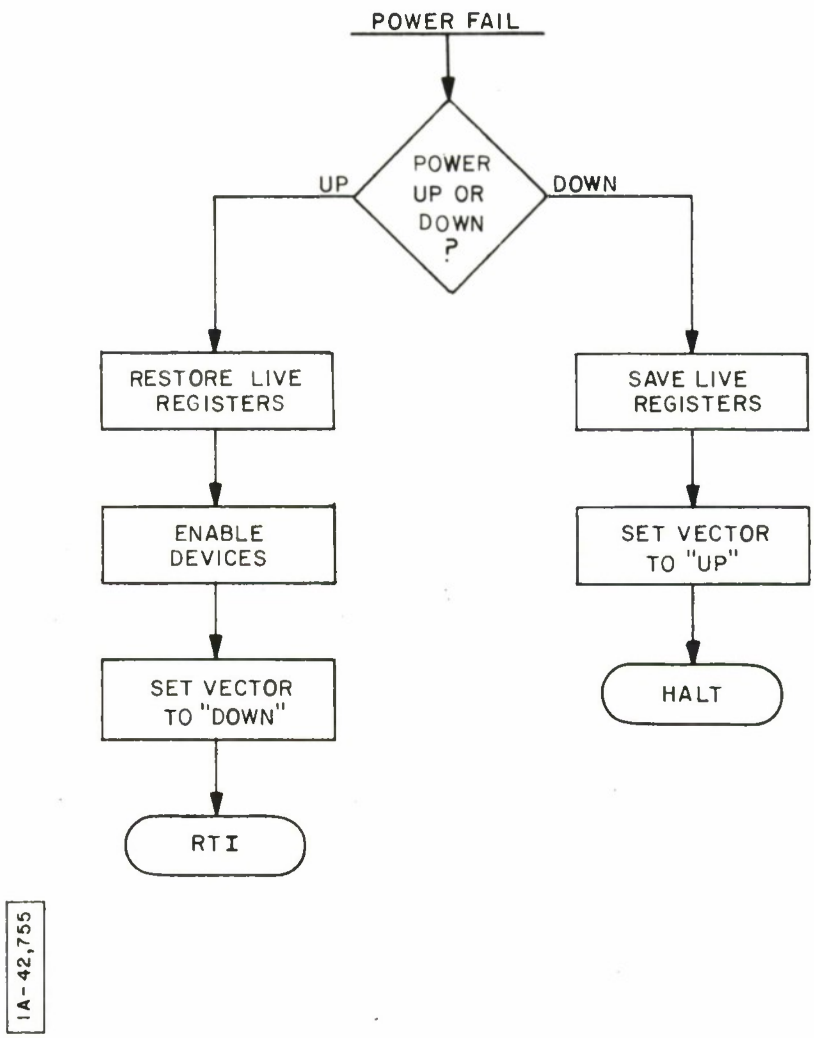


Figure 29 POWER FAIL ROUTINE

The Line Clock routine counts these intervals for one second, at which time it updates a specific core-memory location. This location is, in effect, a register that maintains a count of the number of seconds since program execution began. The register is useful for real-time activities such as time outs. By reading the register at the start of some user or system action, and periodic rereading, a module can determine if the "time-out" time has been exceeded.

4.2.1.8.3 Signaling and Supervision Input (SASI). This routine (Figure 30) accepts data via the S&S controller (Figure 22) from the keypad located at the user terminal and checks it for errors. If the data is valid, it separates the status and keystroke portions and saves them separately. In addition, this routine saves all keystroke information from a user until a complete dialing sequence has been entered. This information is then passed to the Keypad Evaluation module via the Scheduler module.

Upon completion of the sequence the program exits from the Interrupt by executing the RTI instruction, which enables resumption of the normal program flow.

4.2.1.8.4 Signaling and Supervision Output (SASO). This routine (Figure 31) accepts preformatted messages from various modules, checks the external S&S hardware buffer for availability and, if available, sends the message to the buffer for transmission over the S&S subsystem. If the buffer is temporarily full (a rare condition), the routine repeatedly tests the buffer until it is available. Under worst-case conditions, the buffer will only introduce a delay of a few microseconds before it is available for another message.

Upon completion of the sequence the program exits from the Interrupt by executing the RTI instruction, which enables resumption of the normal program flow.

4.2.1.8.5 Keyboard, Teleprinter and Line Printer. These routines are contained within the program coding of the I/O Monitor module itself, as discussed in Section 4.2.1.7.

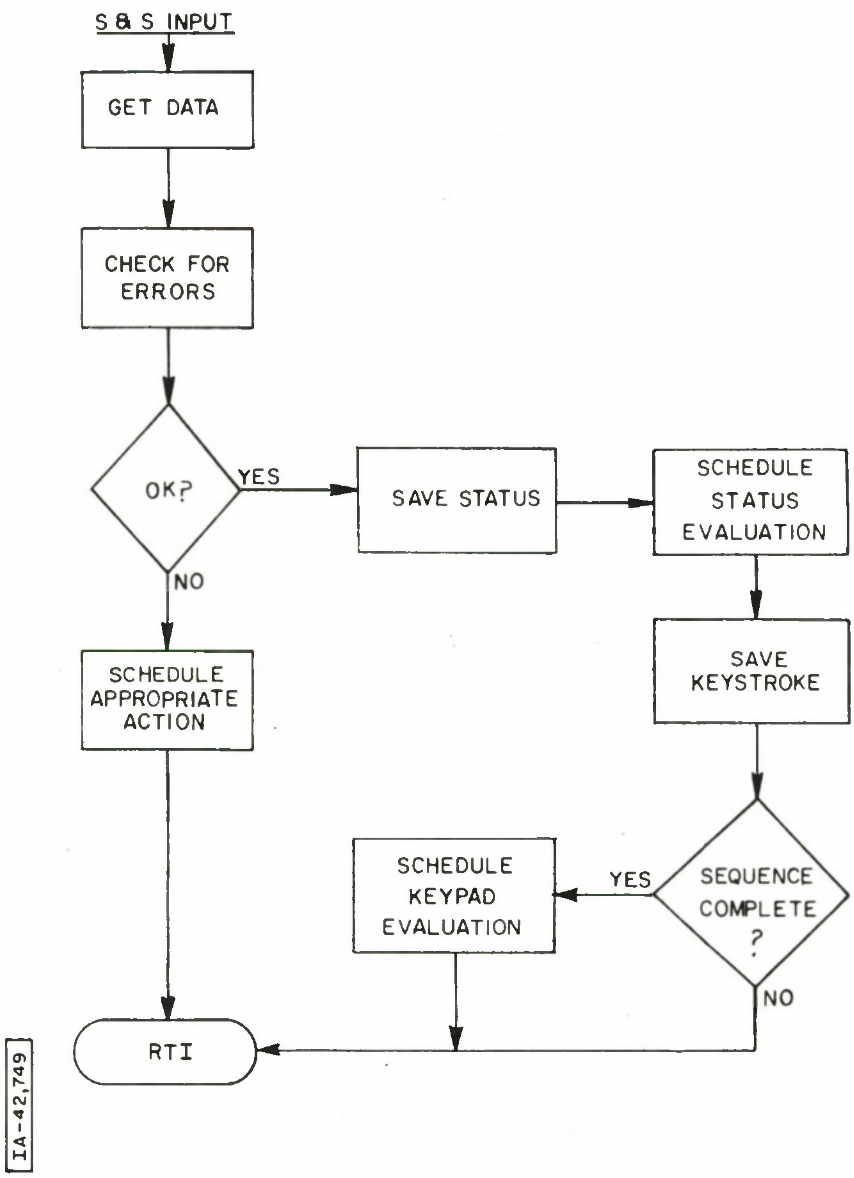
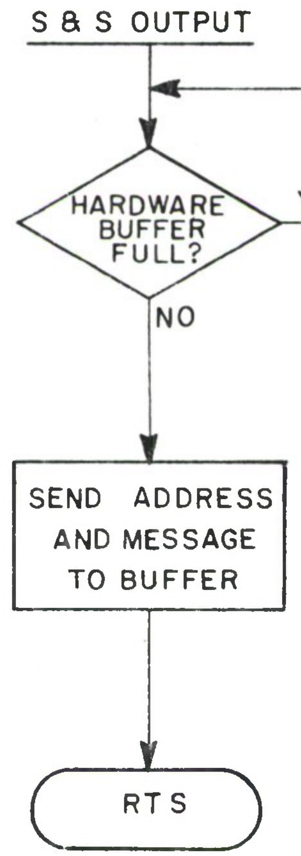


Figure 30 SIGNALING & SUPERVISION INPUT (SASI) ROUTINE



IA-43,573

Figure 31 SIGNALING & SUPERVISION OUTPUT (SASO) ROUTINE

4.2.2 Processor Storage and Timing

An early system design consideration is whether a 16-bit minicomputer has the storage capacity and speed necessary to control the combined data, video and phone traffic. The AFBITS programming efforts on data and video service packages, combined with an investigation of ESS developments by telephone switch manufacturers indicates affirmation, as described in detail in the following sections.

4.2.2.1 NETCOP Storage. The network control programs reside in the PDP-11 memory. These memory storage requirements may be divided into main memory and auxiliary memory. Main memory contains the most often used programs and associated tables and is sometimes referred to as core memory or just "core." Auxiliary memory provides bulk storage and is sometimes called mass memory. Slower but less expensive devices, such as magnetic disks or drums, are normally used for the mass memory function. In the laboratory evaluation configuration, memory storage requirements were small enough to store all programs and data in core memory.

The main memory storage requirements for the NETCOP program consist of two major categories: operating program storage and call-register storage. The operating programs contain the program logic and those tables continuously accessed by the program logic. The call-register storage is used for recording temporary information and consists primarily of dialing-in-progress data and calls-in-progress data.

The operating programs presently coded include Initialization, Dispatch, Scheduler, Keypad Evaluation, Path Selection, Computer Console Operation, Input/Output Monitor, and the Device Interrupt Handlers. Storage for these programs is approximately 4,500 16-bit words. The future extension of these programs to provide all the service package offerings not presently coded, such as high-speed data transmission via digitally tuned frequency synthesizers, also requires 4,500 words, for a total of 9,000 words. In addition the program storage for the Status Reporting, Traffic Recording and Maintenance Diagnostics modules, not presently coded, would double the storage requirement to 18,000 words. It has been found that Maintenance Diagnostics and associated routines may occupy 50 percent of the program storage in large stored-program communications switching processors (Reference 4).

The call-register storage is a function of the number of simultaneous calls being processed. A call-register contains the

digits dialed to establish a call and requires nine words of storage. Assuming a maximum of 2,000 visual display terminals per Air Force Base with 50 percent active at any one time, the call-register storage would be 1,000 users times nine words per call-register, for a quantity of 9,000 words.

The combined storage required for operating programs and call-registers is 27,000 words. This can be handled conveniently by a conventional 16-bit minicomputer with its associated 32K words of storage. A 16-bit minicomputer can access 64K bytes where K = 1024. Since two 8-bit bytes are packed into one 16 bit word, 32K words of storage results.

The auxiliary storage requirements for NETCOP consist primarily of user-data tables that contain the following types of information on each user:

- Service packages allowed,
- Pointers to user's connection links,
- Equipment status such as busy or out-of-service and
- Address (switching-matrix termination) of each equipment.

Fourteen words of data are stored in each user's data table. For 2,000 users, the storage required is 28,000 words. Auxiliary storage such as magnetic disk could be used for the user-data tables. Since even a small disk package can store 1,000,000 words, the same disk could also be used to record traffic usage information for periodic processing later in the day. Alternatively, the user data storage could be provided in "core," which would require a 64,000 word minicomputer. Since most minicomputers have a memory extension feature that permits operating access to 128,000 words of core, the AFBITS network control program could be handled by off-the-shelf minicomputer hardware.

4.2.2.2 NETCOP Timing. For a system with 2,000 visual display terminals, the service request input rate during a typical, busy hour is estimated to be one new request each second. On the average, the duration of a keypad dialing sequence does not exceed 10 seconds. The number of dialing-in sequences, simultaneously processed by the supervision and signaling input (SASI) routine, is 10 per second or one every 100 milliseconds.

To determine the time required to execute SASI, the NETCOP processor's (PDP 11/10) instruction timing must be examined (Reference 5). The instruction execution time for the majority of

the PDP 11/10 minicomputer program instructions is approximately from three to six microseconds, depending on the number of accesses to main memory required by a particular instruction. For example the instruction to move a computer word from one general purpose register to another is coded as: MOV R1, R2 and is executed in 3.1 microseconds because it involves only fetching the instruction itself from main memory. On the other hand, the instruction could use the contents of the registers as pointers to word locations in main memory. Such a MOVE instruction is coded as: MOV (R1), (R2) and involves two separate accesses to main memory after the instruction itself has been fetched. The overall time for this instruction is 6.0 microseconds. (It should be noted that the PDP 11/10 is the slowest and least expensive of the PDP-11 family of minicomputers. The PDP 11/40 is at least twice as fast, with instruction execution times for the two MOVE instructions cited above being 0.9 microseconds and 3.2 microseconds respectively.)

When SASI receives a keystroke "word", it searches its dialing-in-progress tables to determine if the user is in the category of continuing a previous dialing sequence or starting a new one. The search may involve up to 10 tables. Assuming 20 instructions are required to search the table, SASI may execute up to 200 instructions in the search. Since the balance of the SASI program is less than 300 instructions, the worst-case situation that SASI would encounter in processing a keystroke is 500 instructions. Assuming a worst-case time of six microseconds per instruction, the total SASI processing time is three milliseconds. On the average, this leaves 97 milliseconds to continue the processing of requests already "dialed-in" and to monitor existing connections. This is more than ample time to perform the keypad evaluation, path selection, and associated routines.

4.2.2.3 Telephone Storage and Traffic Capacity. Stored program control of telephone switching systems has been under development for many years. The number of actual installations in the Bell System alone increased from 400 in June, 1972 to 700 in January, 1974 (Reference 6). For this reason, the AFBITS NETCOP program coding efforts to date have concentrated on the data and video distribution and control problems. Without doing additional program coding for a laboratory-model telephone switch, modern exchanges such as those developed by the Bell System or the independent Telco suppliers give a good indication of storage requirements for the telephone-switching function. For example, the ITT TCS Switch can accommodate 2,000 subscriber lines using a 16-bit minicomputer stored-program control with 64K of core memory (Reference 7).

The TCS private automatic branch exchange can grow from 2,000 to 6,000 telephone lines with the same minicomputer by increasing the main memory to 128K storage. The 6,000 line exchange can handle 23,000 busy-hour calls. For reliability purposes, the ITT switch is provided with dual processors.

4.2.2.4 Multi-mode Processing. In summary, the multi-mode network control processor requirements can be satisfied by standard 16-bit minicomputer technology. For a 2,000 multi-mode terminal system that combines audio, video and data services, a single processor comparable to the PDP 11/40 minicomputer with 128K core memory is adequate. For reliability and maintenance-training purposes, a backup minicomputer would also be necessary. For very large Air Force Bases with 10,000 AFBITS subscribers (2,000 multi-mode terminals and 8,000 telephones) a multiprocessor configuration could effectively control the integrated data, video and phone services required.

4.3 Program Routines

This section describes, in detail, the routines actually coded by AFBITS project programmers for use in the laboratory evaluation configuration.

4.3.1 System Initializer (INIT)

When the NETCOP program is activated, it causes a direct transfer to INIT, whose role is initialization of the NETCOP software. The INIT makes calls to other routines that require setting various counters, switches, and addresses to zero or other starting values and also transfers control to the system dispatcher (DISP) when initialization is completed.

4.3.2 Signaling & Supervision Input (SASI)

SASI is the device interrupt handler that services interrupts transmitted from an asynchronous Time Division Multiplex (TDM) hardware device controller. The interrupt signals SASI that a dialing-in-progress function is in effect and to fetch two 16-bit input words from the TDM's FIFO (First-In/ First-Out) buffer storage area (i.e., the contents of the upstream buffer).

The resulting digital data contains the subscriber's (user) address in the first word and keystroke/status information in the second word. Upon inspection, SASI determines if the input is of the keystroke (user) type or of the status (e.g., system or program error mode or both) type. Keystroke types of input are stored in a pooled buffer for future evaluation & processing by scheduling the program Keypad to run after a complete user input sequence is detected by SASI. All status-type input conditions have not been implemented in SASI for the laboratory evaluation configuration. Presently, SASI only sets an error flag for Keypad when a system hardware error has been detected.

The program logic for SASI is described in the following paragraphs. Additional details on the program logic are available from the project programmer's detailed flowcharts and the associated program listing (coding sheets). A condensed version of the programmer's detailed flowcharts is provided in Appendix B.

Whenever the SASI handler receives an interrupt from the TDM device controller, it initially saves all applicable general-purpose registers and services the interrupt prior to fetching two 16-bit computer input words from the TDM upstream buffer register. Word 1

contains the user (i.e., calling) source address. Word 2, associated with the user address, contains keystroke/status information and will henceforth be referred to as a "message."

A check is then made in a table internal to SASI to determine if the data received is the beginning or the continuation of an incoming message sequence. This table is structured to maintain a 2-word pair slot for each input that is still in progress (i.e., not yet complete). The first word in the table contains the user address proper and the second word contains an associated node address. Note that a node acts as the working storage area for user input. The node is obtained via a common-storage subroutine called POOL (Section 4.3.7).

If the message is the very first message that constitutes an input sequence (this is ascertained by the absence of the user address in the table), the handler will flag the user address as "new", i.e., set an indicator to call POOL for later use. The user address is then saved in a register prior to inspecting the message word. If the message is found not to contain any user data, i.e., neither Bit 4 nor Bit 5 is set, a "system error" mode condition exists. (This occurs when the signaling and supervision hardware controller detects a parity error in a particular keypad transmission. It uses a zero in bits 4 and 5, the Change Indicator bits of Figure 11, to notify the network control processor of the error). A test is then made to determine how to further process said message . The test proceeds as follows:

(1) If the user address is "new," the message is cleared and SASI forces it to be a Call Modification Request and sets the appropriate code in the keystroke field of the message word.

(2) If the user address is not "new", the digit control count is decremented and an error bit is set in the appropriate word field of the user-node block prior to making a normal exit from SASI via a Return-From-Interrupt (RTI) instruction. Note that a non-new user has already been assigned a node from a previous input and that the detected error condition is subsequently treated by the keypad evaluation routine, 'KEYPAD.'

If the message contains status information, the SCHEDULER routine is called. The SCHEDULER records that a status evaluation

is to be performed and then returns program control back to SASI so it can finish servicing the interrupt.

The handler's next step is to test bit 4 of the message. If that bit is not set, it means the message did not contain keystroke data; hence, it is ignored and an RTI is executed. If bit 4 is set, the keystroke will be processed in one of two ways, depending on whether or not the user address is "new."

For a new input, SASI first schedules the READY lamp to light at the user work station. The "new" flag is cleared, and the keystroke part of the message is then inspected to determine its function. If it represents a CLEAR function, it is ignored and SASI does an RTI. Otherwise, a call register node is obtained from POOL and the user's system address/node's memory location information is stored in the handler's table. Further inspection is then carried out in the following manner:

(1) Should the message contain a REDIAL keystroke, the redial and dialing complete bits are set and the digit count is set equal to one in the appropriate user node positions; the WAIT light at the user work station is scheduled to be turned on; KEYPAD is scheduled; and the user address is removed from the handler's table just prior to an RTI.

(2) If the message contains a service keystroke, the keystroke information is extracted and stored in the user node; also, the service bit and updated word count value are set in the node before the RTI is executed.

(3) If the message contains a call modification keystroke, the call modification bit is set in the user node along with its appropriate word count value, followed by an RTI.

If the message contains none of the above three keystrokes, it is granted to be the first digit of a four-digit input request. It, therefore, is extracted and deposited in the user node in a position relative to the digit count. The digit count is then updated prior to an RTI.

Input which is not of type "new" is processed in the following manner:

(1) Upon inspection, if the message contains a "DISCONNECT" keystroke, the disconnect and dialing complete bits are set in the user node, the WAIT light at the user keypad is turned on, KEYPAD is scheduled, and the user address in the handler's table is removed prior to an RTI.

(2) Should the message contain a CLEAR keystroke, POOL is called in order to release the user-node block, and the user address is cleared from the table before the RTI. If the message contains neither of these two keystrokes, the keystroke is extracted and deposited in the user node relative to the current digit count value. The digit count is then updated and a test is performed to see if the input request sequence is or is not completed. If incomplete, SASI will RTI. For a complete input request sequence, the dialing-complete bit is set, the WAIT light at the user work station is turned on, KEYPAD is scheduled, and the user address is removed from the handler's table prior to an RTI.

4.3.3 <u>Signaling and Supervision Output (SASO)</u>

The signaling and supervision output program, SASO, is a relatively short subroutine that is called by other NETCOP programs desiring to send a command or control word to external equipments such as the switching-control units located at the hubs.

The routine calling SASO passes two computer words to it that contain the external device's system address and the single-word command message for the external device.

SASO transfers this information to the downstream register of the signaling and supervision hardware controller's First-In/First-Out (FIFO) buffer, provided the controller's buffer is capable of accepting the two computer words. If the controller is temporarily full, SASO continues to test the register until it is available.

The SASO program logic is quite simple as illustrated in Figure 31.

4.3.4 Keypad Evaluation

The keypad evaluation routine, KEYPAD, processes call registers that are created by the Signaling and Supervision Input (SASI) routine from keystrokes sent from users' keypads. The call registers contain the user's address, the keystrokes that were sent, and some status information. The call register contains a complete dialing sequence of from one (DISCONNECT or REDIAL) to nine (modify service package) keystrokes. The functions of KEYPAD are to:

- Check the validity of the calling sequence,
- Check the status of equipments to be connected,
- Select common equipment to be used,
- Find a path between equipments (using the PATH subroutine),
- Connect or disconnect equipments (using PATH),
- Book keep connections so that the processor has a record of all calls in progress and
- Light the ready, busy, or error light on the calling party's keypad.

Keypad evaluation always results in one of the following:

- The user's READY light being turned on, meaning that the proper connections or disconnections were made.
- The ERROR light being turned on, meaning that there has been an error in the call-request and it cannot be processed.
- Or the BUSY light being lit, meaning that no connections or disconnections have been made, but a record is being kept of the calling sequence that the user can reactivate by keying REDIAL.

KEYPAD turns on an ERROR light for the reasons cited below:

- Time out,
- Hardware error,
- Invalid keystroke, such as a non-numeric digit in the address field,
- Unassigned directory number, service package number or call modification number,
- Service package, which user is not allowed to use.
- Called party, who doesn't have the equipment necessary to receive a call,
- Invalid attempt to modify a service package, such

as dropping a party not previously added by a call modification and

- REDIAL which was not immediately preceded by a busy signal.

KEYPAD turns on a BUSY light if any equipment necessary for a service package is either in use or out of service. The equipment can be at the calling party's work station, at the called party's work station, or it can be common equipment. Regardless of the cause of the busy condition, KEYPAD saves the call-register for REDIAL until another call-register is processed for the same user which contains something other than REDIAL. DISCONNECT does not normally disable the REDIAL option, but a Service Package request or a Call Modification will disable REDIAL. An incoming call has no effect on REDIAL.

Service package requests, keyed as S XX XXXX, result in a series of connections between equipment at the calling party's work station, equipment at the called party's work station (if there is a called party), and common equipment necessary for the service package. The sequence of connections resulting from a service package request is defined in the service package format table. Each connection in the table is represented by a general description of a "from" and a "to" terminal. "From" and "to" terminals have two parameters.

The first parameter consists of a code indicating one of the following:

- Calling party
- Called party
- Or Common Equipment Type.

The Second parameter indicates one of the following:

- Audio input terminal,
- Audio output terminal,
- Video input terminal,
- Video output terminal,
- High-speed data input terminal,
- High-speed data output terminal,
- Low-speed data input terminal, or
- Low-speed data output terminal.

An example of a service package is microfiche service, where the user's keyboard is used to remotely control a microfiche unit and

the video output of the microfiche unit is sent to the user's monitor. For this service package, two connections will be defined in the service package format table. The first is a connection from the calling party's low-speed output terminal (his keyboard) to the microfiche control unit's low-speed input terminal. The second is a connection from the microfiche's video output terminal to the calling party's video input terminal (his monitor). The service package could be set up in one of two ways. The microfiche unit could be given an address that the user would have to key. It would be treated by the software as a "called party." The user could then select the particular unit he wanted to use. In the second method the unit would not be explicitly addressed, but an "equipment type" (a port on a microfiche unit) would be implied by the service package request. The user, by keying "S," the service package number, and any four digits, would be connected to one of several units depending on their availability. With this service package definition the microfiche unit is treated as "common equipment" and is referred to in the service package format table as an "equipment type."

For the first set up the connections would be:

From: calling party's low-speed output terminal
To: called party's low-speed input terminal.

From: called party's video output terminal
To: calling party's video input terminal.

For the second set up the connections would be:

From: calling party's low-speed output terminal
To: equipment type "#1" low-speed input terminal.

From: equipment type "#1" video output terminal
To: calling party's video input terminal.

where equipment type "#1" refers to a group of ports on a microfiche unit which could be used interchangeably for the desired service.

Call modification is a method of changing the configuration of connections between equipments in a service package. Call modification is legal only after a service package (or phone call) has been established. It is used primarily for broadcasting audio, video or data output. Call modification is keyed: M X S XX XXXX. This results in one connection or disconnection between equipments belonging to the calling party, common equipment that is already in

use in the service package, and equipment belonging to the new called party. The call mod number, the number keyed after the "M," determines what equipments are involved and whether they should be connected or disconnected. Equipments may be disconnected by a call mod only if they have been connected by a call mod. The setup of the service package format table determines whether or not a service package can be modified, and if so, what call mod numbers are used for what connections or disconnections.

The service package format table contains a field for "alternate service package." This field will normally be zero, but may also contain the number of another service package that the program will automatically try to set up for the user if unavailability of equipment prevents him from getting the service package he keyed. The alternate service package may be a directory-listed service package that the user could have requested or one that is illegal for the user to key and is known only to the computer program. An example of a use for the alternate service package option is for low-speed data service. Low-speed data service can be set up to connect the calling party's low-speed output terminal to the called party's low-speed input terminal and the called party's low-speed output terminal to the calling party's input terminal. If, however, there are work stations that can send but cannot accept low-speed data, they can be allowed to use the "low-speed data" service package, but they will be given an alternate service package which contains only a one-way connection, from the calling party's low-speed output device to the called party's low-speed input device.

A user may initiate any number of service packages simultaneously, up to the limit of his equipment. He can initiate one, initiate another, then modify or disconnect the first without affecting the second. He can also be the "called party" of a service package initiated at another work station without it having any effect on his own calls. A record is kept in the computer of all calls in progress, including the original service package request and any additional connections that were made by way of call modification. Information about a call is tied to the call initiator; only he can either modify or disconnect the call. He has the option to:

(1) Drop (by call modification) a connection which he previously added by modification.

(2) Disconnect an entire service package, including modifications, by keying M S XX XXXX. (The last four digits are ignored, and this sequence is interpreted as "Delete Service Package XX."),or

(3) Hit the DISCONNECT key, that disconnects all service packages initiated at his keypad.

At the time of disconnect, traffic data will be recorded for subsequent traffic analysis. All other memory of the call is deleted and disconnected equipments are made available for other calls.

Up to this point in the KEYPAD description, connections between equipments have been discussed as though they were single physical connections. Actually a "connection" as represented in the service package format table may result in several physical connections, depending on the relative location of the equipments to be connected and the characteristics of the signal to be sent. If, for instance, the calling and called parties are at different hubs, the connection may have to include a trunk channel, or special equipment may be necessary to modify a signal in order to make the input terminal compatible with the output terminal. This additional equipment is not included in the service package format table. The selection of this equipment, the formatting and outputting of physical connect and disconnect commands and the bookkeeping associated with path selection is accomplished by the PATH subroutine (Section 4.3.5). KEYPAD tells PATH to connect equipment A to equipment B. PATH determines what, if any, intervening equipment is necessary. It selects this equipment from a pool of such equipment and records its selection. PATH then issues commands to make the necessary physical connections. On disconnect, KEYPAD again calls PATH, which decides what physical disconnections to effect.

Input to one pass of KEYPAD is one call-register from the signaling and supervision input routine. This call-register is modified somewhat by KEYPAD. (The service package number and address of the called party are converted to binary and certain other fields are changed.) After KEYPAD has been executed the call-register is either destroyed (if it contained redial, disconnect, an error, or a delete type of call modification) or it is saved as a record of the call. When a call-register is saved it is associated with the calling party. Each user is described by a "user data table," that contains both permanent and variable data describing the user work station and the status of its equipment. In the table, there are pointers to the beginning and end of a chain of

variable data used to describe calls in progress. This chain consists of call-registers that are linked together by pointers in the order in which they were processed by KEYPAD. They are either service package requests or requests to modify service packages by addition. Normally, these call-registers represent completed calls, but the last link may represent an incomplete call where the call-register is being saved for REDIAL.

This chain of user requests is kept so that call modifications and disconnections can be made properly. If a service package is deleted or a party is dropped from a call, the call-register that requested the service package or added the party to the call is removed from the user's chain and destroyed. A Disconnect causes the entire chain to be processed and then destroyed; service packages are deleted one at a time, starting with call modifications. As each call-register is processed, it is removed from the chain until the last service package has been deleted.

The user chains are only one aspect of the bookkeeping that KEYPAD performs. Another bookkeeping function is to keep track of available, in-use and out-of-service equipment. Equipment that is at a user's work station has its status recorded in the user data table. In a large system, user data may be stored on an auxiliary storage device such as a disk. In the laboratory evaluation system, user data is kept in "core," but logic is included for the accessing and storing of these tables.

The common equipment that is specified in the service package format table and controlled by KEYPAD is described in the common equipment tables. An in/out of service bit is provided for each equipment. When common equipment is in use the address of the calling party and the service package number are stored in the common equipment table. These two fields are cleared when the equipment is disconnected. Common equipment tables are always kept in "core."

NETCOP allows for restricting the use of service packages to certain users by including in the user data table a field of indicator bits indicating which service packages the user is allowed to use. KEYPAD checks this field on a service package request and activates the Keypad ERROR light if an illegal service package has been requested.

A user category field is also provided. This field is intended primarily to ensure that users at a hub access common equipment located at the same hub. The common equipment selection routine

checks the user category before assigning common equipment to a call. Common equipment is classmarked with the user categories that are allowed to use it. By assigning user categories according to hubs, a user can be prevented from using equipment at other hubs, and thus will automatically get equipment at his own hub or else will receive a busy signal. User categories can also be used to allow only authorized subscribers to access particular files.

The Keypad Evaluation routine makes extensive use of cross comparison of data tables, and this type of program logic is sometimes called "table-driven logic." A detailed description of each table used in KEYPAD follows.

4.3.4.1 Call Register Tables. The call-register (Figure 32) is KEYPAD's only input. It represents one complete dialing sequence. It contains the calling party's address in binary and the keyed digits, that appear in the call register either as four-bit codes as they were received from the keypad or as indicator bits that were set by the SASI routine. The call-register is located within the storage pool. It is accessed by way of a pointer within the KEYPAD program that is set by SCHEDULER.

A minimum of validity checking has been done before the call-register is turned over to KEYPAD for evaluation. If the time-out, hardware error, redial or disconnect bits are set, no further checking is necessary because any one of these bits represents a complete dialing sequence. If the service bit is set, "S" was the first keyed digit and the next six digits were stored, unchecked, in words 5, 6 and 7 of the call register. These digits should contain the service package number and the address of the called party. If the call mod bit is set, "M" was the first keyed digit and the next eight digits were stored in words 4, 5, 6 and 7 of the call register. These digits should contain the call modification number, the code for "S," the service package number and the address of the called party.

4.3.4.2 User Data Table. Data on each user work station is maintained in a User Data Table (UDTBL), Figure 33. A "user" is a keypad with its associated equipment. User equipments share the directory address of the keypad and are differentiated from each other by their terminal type which is audio input or output, low-speed data input or output, high-speed data input or output, or video input or output. Each user (or work station) can have one each of the eight terminal types. NETCOP does not recognize differences in equipment within a terminal type. Compatibility of

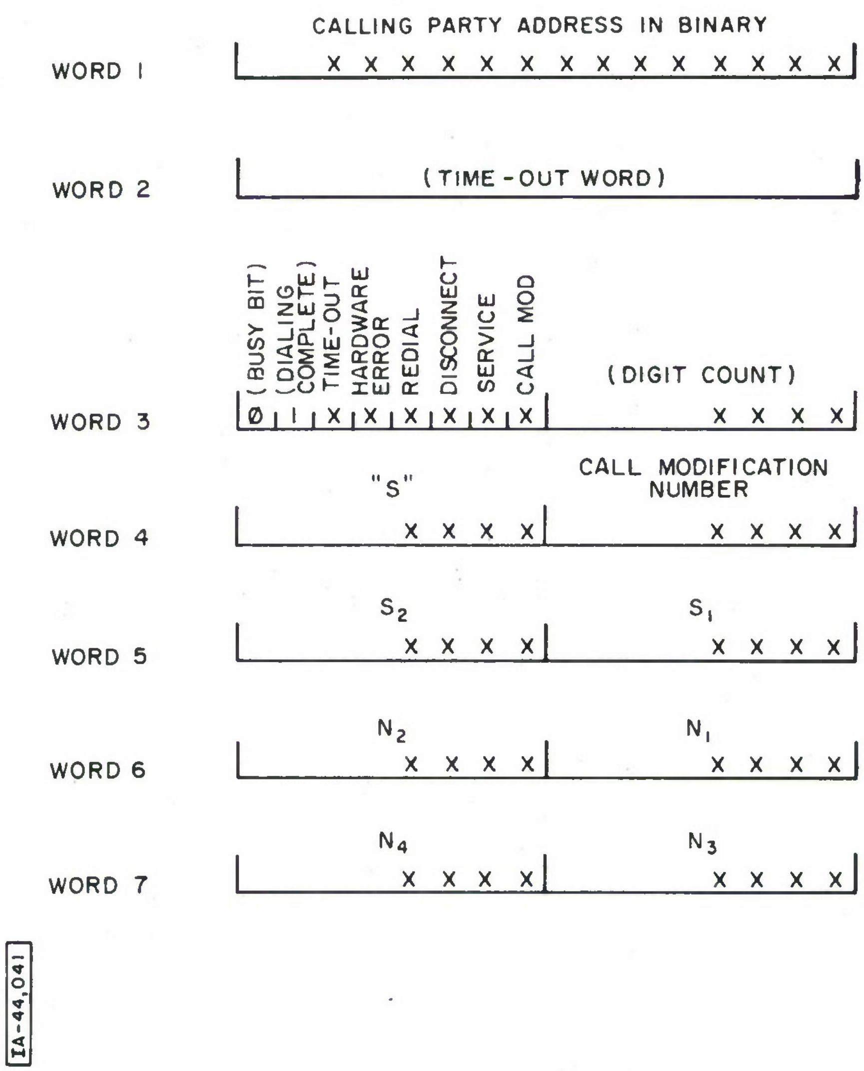
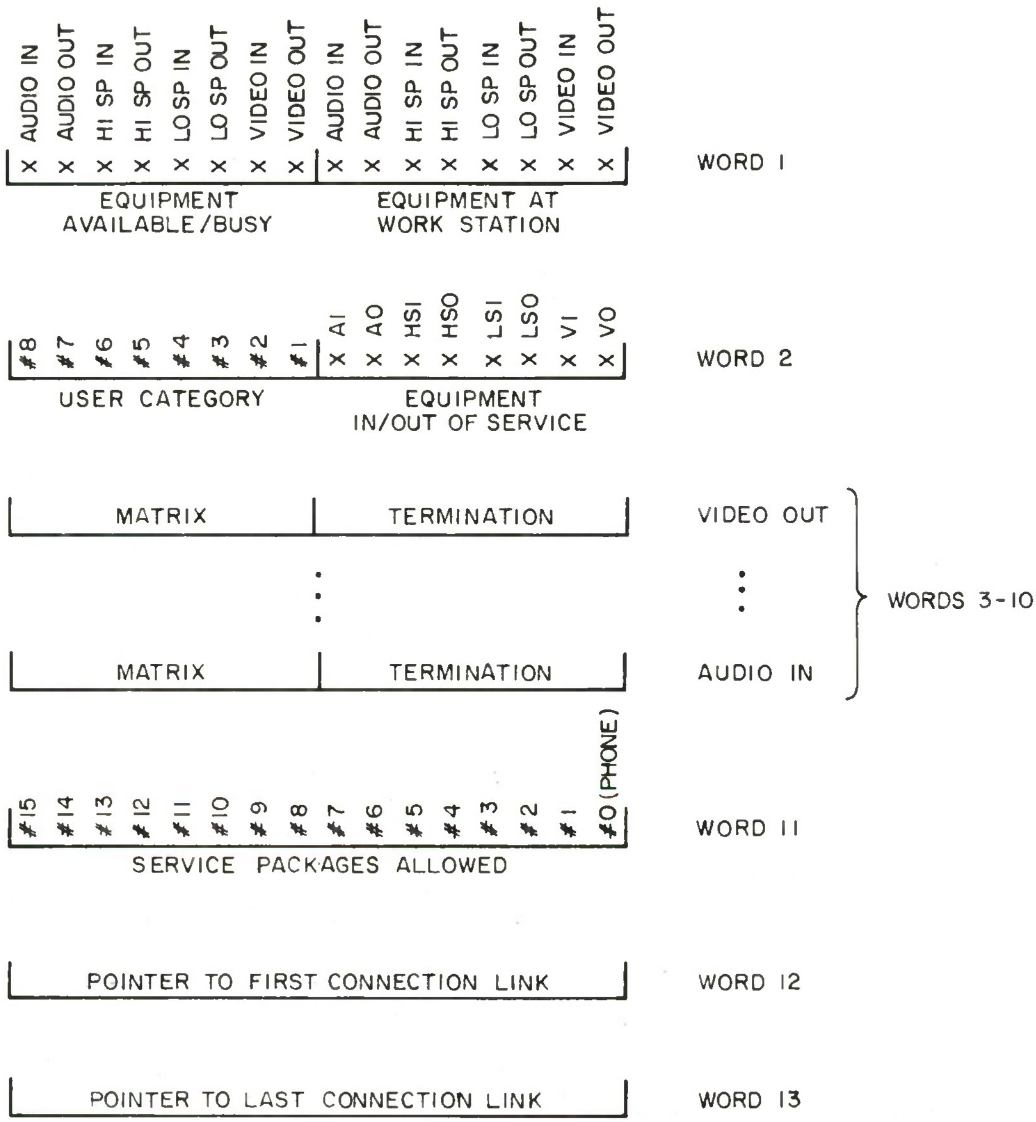


Figure 32 KEYPAD EVALUATION INPUT - CALL REGISTER



IA-44,037

NOTE : ONE ENTRY PER SUBSCRIBER, ACCESSED AS A FUNCTION OF THE KEYPAD ADDRESS

Figure 33 USER DATA TABLE (UDTBL)

equipments that will communicate with each other is the responsibility of the user.

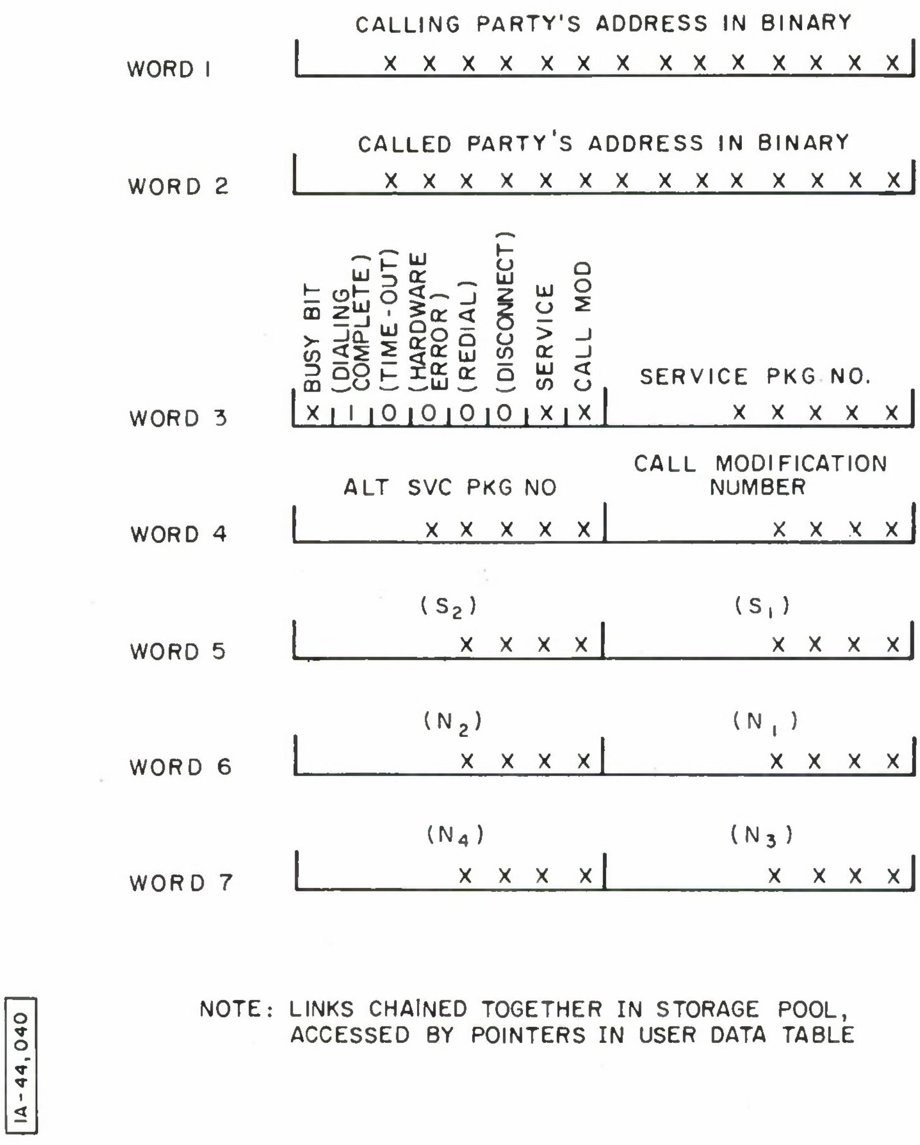
For each of the eight terminal types there are three indicator bits: one bit tells whether or not the equipment is present at the work station, another is an in-or out-of-service indicator, and the third is an available or busy bit. The busy bit is set for an equipment involved in a service package setup by NETCOP. If an equipment is busy off-line, NETCOP has no way of knowing its status and regards the equipment as available.

One word is allowed in the user data table for the address of each of the eight terminal types. In the present program, all equipments are connected through matrix switches. The address is the matrix number and the termination on the matrix. In future systems, the address field may have another meaning. Indicator bits are assigned, one per service package, for service packages allowed. This is an optional feature used to prevent an unauthorized subscriber from using certain service packages.

There is a user category field that is used in conjunction with the user category field in the common equipment tables to associate particular pieces of common equipment with classes of subscribers that may use them. This also is an optional feature. Its primary function is to ensure that a subscriber is connected to the equipment closest to his work station. It can also prevent unauthorized subscribers from obtaining access to certain files.

Records are kept in the storage pool of the dialing sequences used to set up all calls in progress. The information is stored in a form similar to the call-register that was input to KEYPAD. The format of this modified call-register is described as the "user's connection link." (See Figure 34.) Connection links resulting from calls made by a subscriber are chained together in the order in which they were processed as call-registers. Pointers to the first and last links of this chain are stored in the user data table of the subscriber who initiated the calls. Absence of a user chain is indicated by minus one in both pointers.

The chain of user connection links is a logical extension of the user data table. The information is kept in the storage pool to save space in the user data tables. A link is formed in a subscriber's chain when he makes a service package request or modifies a service package by addition. Links are removed from the



NOTE: LINKS CHAINED TOGETHER IN STORAGE POOL, ACCESSED BY POINTERS IN USER DATA TABLE

Figure 34 USER'S CONNECTION LINK

chain when a service package is deleted, a service package is modified by dropping a connection, or the subscriber keys "disconnect."

Links normally represent completed connections; however, whenever a subscriber dials and receives a busy signal, a link is saved for redial with its busy bit set. This link is always the last link in a user's chain. If the subscriber keys REDIAL, the link is treated as an incoming call register. If the subscriber keys another service package or modification, the link is deleted.

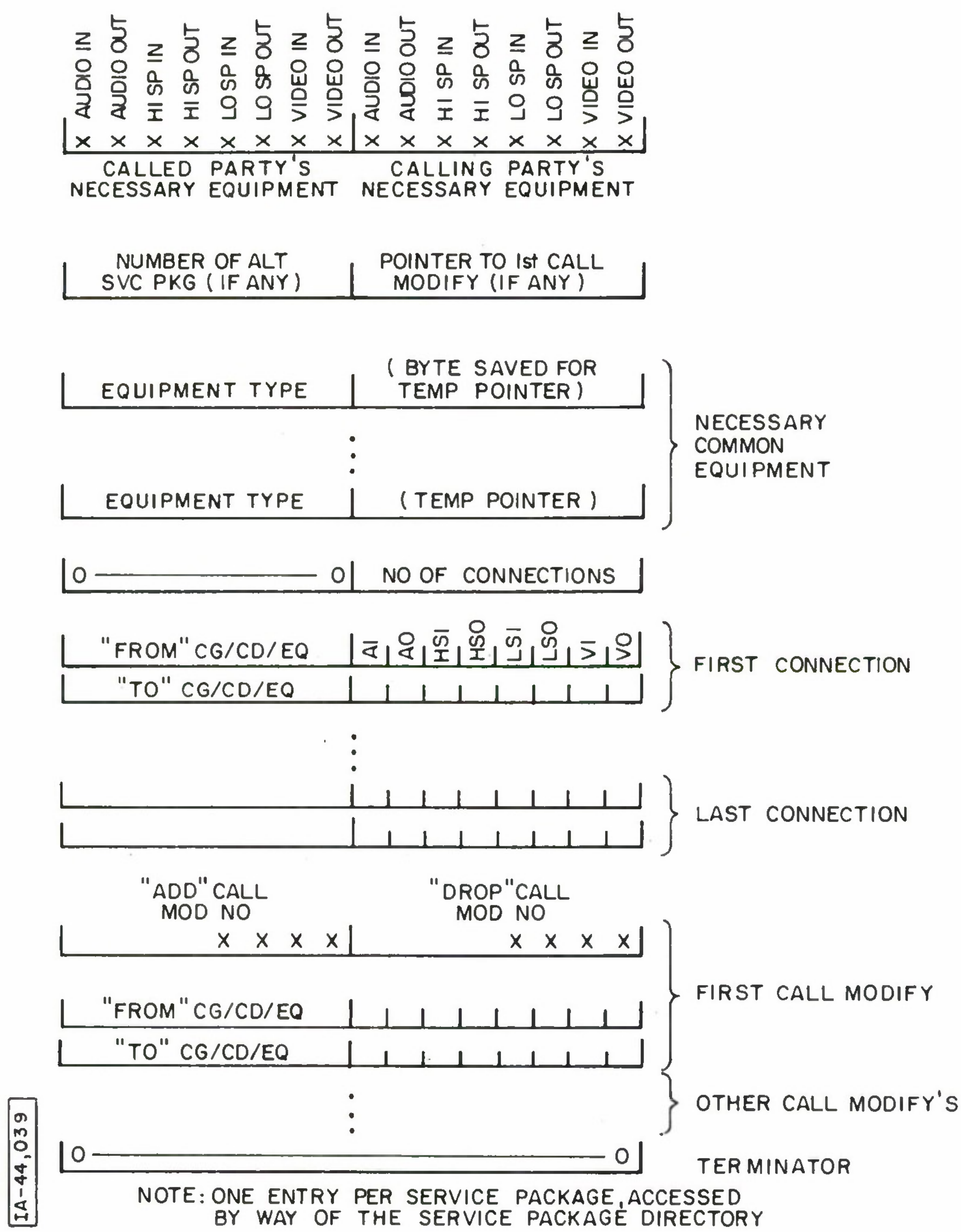
4.3.4.3 Service Package Format Table (SPFT). The Service Package Format Table, SPFT, (Figure 35) describes the service packages that are implemented for the installation. It is accessed by the service package number via the service package directory. It contains:

(1) The connections to be made between the calling party's equipment and common equipment.

(2) The call modification numbers and the single connection or disconnection that results from each.

(3) The number of an alternate service package that should automatically be used if unavailability of equipment prevents a user from getting the service package requested.

The first two fields, the calling party's necessary equipment and the called party's necessary equipment, are used to make quick checks on the availability of equipment. The equipment type numbers of necessary common equipment are listed in the service package format table for the same reason. No connections are made for a subscriber requesting a service package unless all user equipment (calling and called) and common equipment is available. If any equipment is unavailable, the alternate service package is tried (if there is one); if not, the calling party is given a busy signal.



NOTE: ONE ENTRY PER SERVICE PACKAGE, ACCESSED BY WAY OF THE SERVICE PACKAGE DIRECTORY

Figure 35 SERVICE PACKAGE FORMAT TABLE (SPFT)

Connections described in the service package format table are of the form:

```
                    H H L L
                A A S S S S V
CG/CD/EQ        I O I O I O I O    "FROM" TERMINAL


                    H H L L
                A A S S S S V V
CG/CD/EQ        I O I O I O I O    "TO" TERMINAL
```

```
where CG = 200 (octal)
      CD = 201 (octal)
      EQ = the equipment type number in binary
```

One bit is set in the indicator bit portion of each word to show the terminal types to be connected. Assuming that a framegrabber is equipment type 2, the connection between the framegrabber´s video output terminal and the calling party´s monitor is shown below:

```
     EQ #2        Video Output
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1


  Calling Pty     Video Input
1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
```

A single connection in the service package format table may result in more than one physical connection; if, for example, the equipments to be connected terminate on different matrices or are incompatible and require intervening equipment. Trunk channels and other equipment necessary to complete a connection for a particular configuration of equipment are not included in the common equipment tables referenced by KEYPAD.

Service packages that can be modified have a pointer to the call modification portion of the service package format table. If this pointer is zero, no modifications are allowed. Call modifications are defined in pairs. A single from-to connection is used to define an add and a drop modification. The call mod number determines whether a connection or a disconnection should be made. Only one connection is allowed for a call modification.

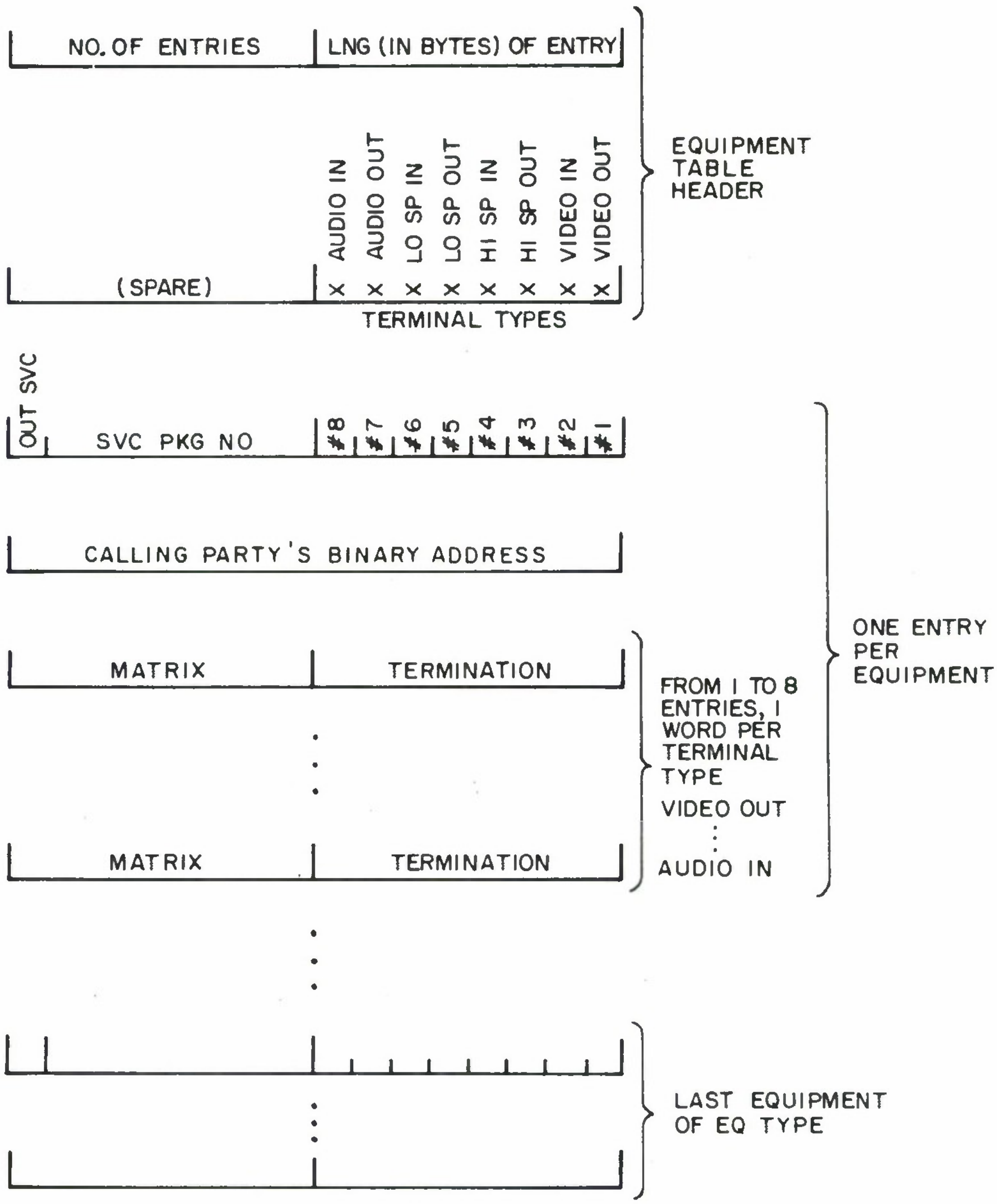
The alternate service package feature is optional; a zero in the alternate service package field means there is no alternate service package. If a user is subsequently given an alternate service package in place of the service package he dialed, modifications will be governed by the format table of the alternate service package.

4.3.4.4 Common Equipment Tables. Common equipment is equipment, such as a microfiche unit, that is not explicitly addressed in the dialing sequence, but is necessary for a service package. Groups of similar common equipment are given equipment type numbers. Equipments of a given equipment type number are functionally identical and can be used interchangeably with other equipments of the same type. Each equipment type is described in a common equipment table, EQTBL, that is referenced by way of the common equipment directory (Figure 36).

The equipment table header contains indicator bits showing the terminal types (low speed data in, video out, etc.,) present on the equipments. Most equipments will have either one or two terminal types, but provision is made for all eight. The NETCOP program does not allow for more than one terminal of the same type.

The common equipment table contains the matrix address of each terminal for each equipment. It also contains indicator bits for acceptable user categories. Assigning subscribers to user categories and allowing only certain categories to use particular equipments is a simple method for subdividing similar equipment by hubs. Space is allowed in the equipment tables for the binary address of the calling party and the service package number. These fields are set when the equipment is selected for a service package and are set to zero when the equipment is released.

Additional details on the program logic are shown in a condensed version of the programmer's detailed flowcharts, Appendix C.



IA-44,038

NOTE: ONE TABLE PER EQUIPMENT TYPE, ACCESSED BY WAY OF COMMON EQUIPMENT DIRECTORY

Figure 36 COMMON EQUIPMENT TABLES (EQTBL)

4.3.5 Path Selection

The function of the PATH subroutine is to select intermatrix paths and formulate messages for the matrix switch control units. As illustrated in Figure 37, the PATH subroutine may be called by the System Initialization routine or by the Keypad Evaluation routine. When it is called by the System Initialization routine, its function is to send a "clear all cross points" message to each switch control unit in the system. When it is called by the Keypad Evaluation routine, its function is to find an end-to-end path from the calling party to the called party. If the calling party and the called party are on the same switch, it must determine the exact crosspoint(s) to be closed. If the calling and called party are terminated on different switches, it must also determine the trunk(s) to use between the switches.

After it has determined the path, it puts this information in the switch control unit format. The PATH routine then calls the SASO routine and provides it with the Switch Control Unit address and a command message. The SASO routine interfaces with the hardware S&S TDM controller that is the actual physical signaling path to each switch control unit. After the Path routine has sent the message via the SASO routine, it then returns control back to the Keypad Evaluation routine via the Return From Subroutine (RTS) software instruction.

4.3.5.1 Message Formatting. The relationship between the processing of the internal minicomputer words and the resulting hardware downstream character transmission for the routines described above is illustrated in Figure 38. The Keypad Evaluation routine hands-off three words to the path selection subroutine. Word number one contains the switch command, that may be either a disconnect or a connect-type command as identified by bits 6 and 7. Word number two contains calling party information and identifies the switching matrix and input termination on which the calling party is terminated, hence the use of the "From" termination designation. (The "From" termination is also identified as the X input.) Word number three contains information on the destination party, i.e., the switching matrix and output termination for the called party are identified, hence the "To" termination designation. (The "To" termination is also identified as the Y output.) The bit structure of each word is indicated with the low-order bit being identified as zero. Correspondingly, the low-order byte and the high-order byte of the the two-byte word is also identified.

IA – 44,416

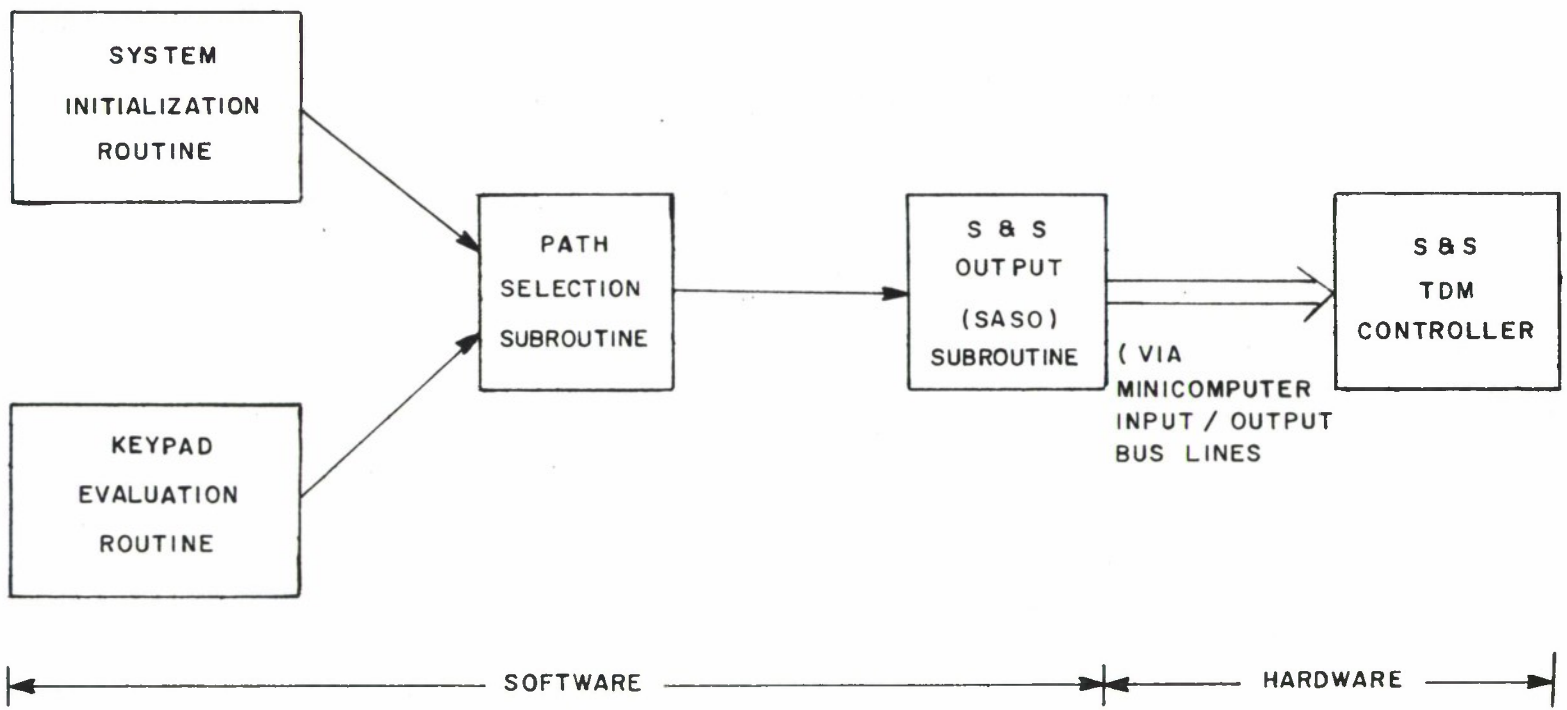


Figure 37 PATH INFORMATION FLOW

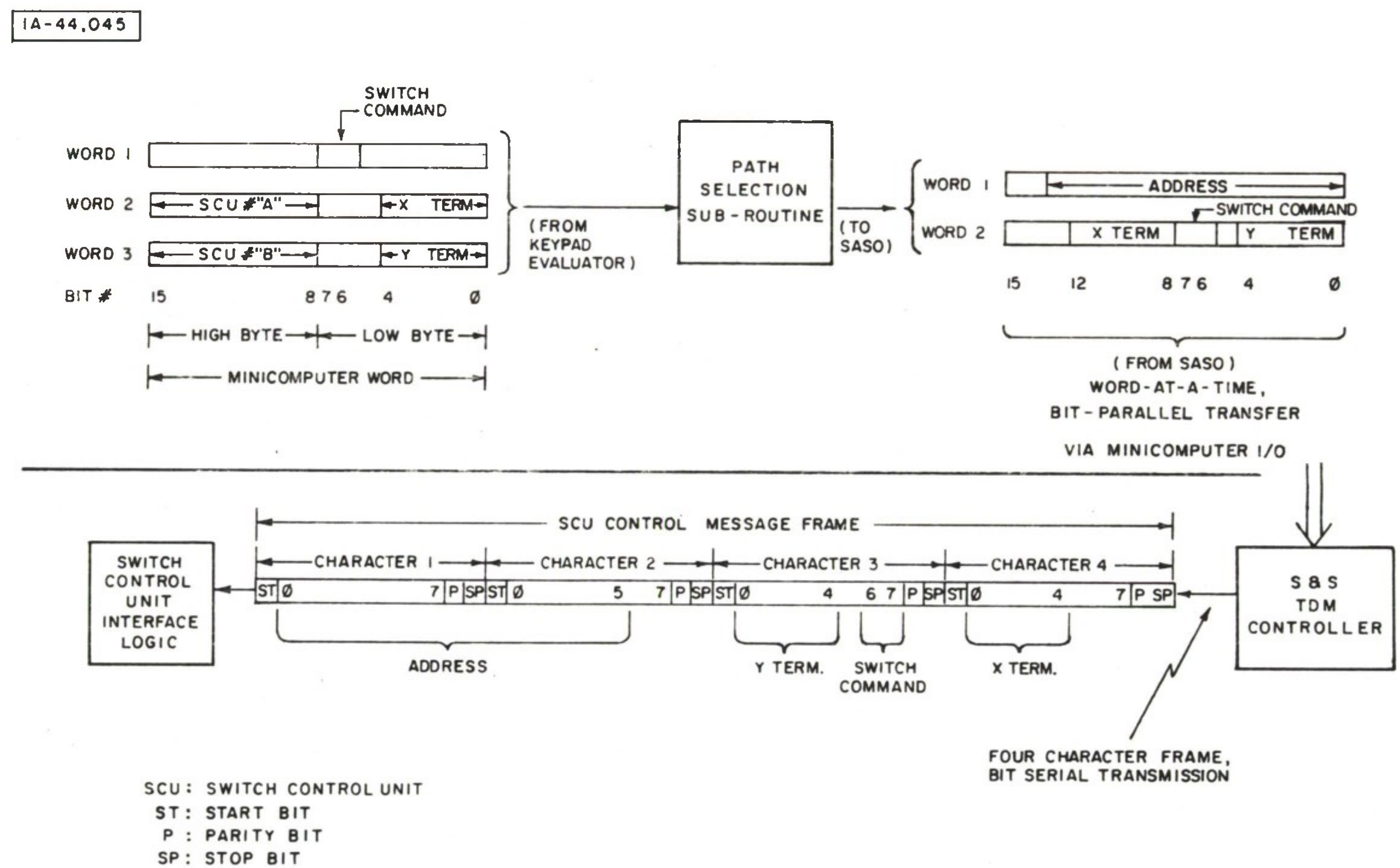


Figure 38 PATH SELECTION MESSAGE FORMATTING

PATH operates on this three-word combination, and in conjunction with its own internal trunk tables, determines the route and the switching-matrix crosspoints to be closed. After it has made this determination, it passes off one or more messages to the appropriate switch control units so that the end-to-end system connections can be established. It does this by a series of two-word messages, that the software routine SASO uses to send to the hardware TDM controller.

The first word contains the system address. It is the signaling and supervision address that the TDM controller will send downstream. The second word contains the message for the switch control unit in a format such that it will know which crosspoint to close. SASO transfers the words, one at a time, bit-parallel to the signaling and supervision TDM controller. When the TDM controller has received the two words, it sends them down as a four-character serial bit stream. The bit arrangement for each character is shown in Figure 38; the address arrives at the switch control unit first, followed by the Y termination/command character, followed by the X termination character. The handoff by SASO via the S&S controller and the resulting downstream character transmission is also typical of the process that is used to send information back to the keypad.

4.3.5.2 Program Components. PATH has three program components: the System Initialization routine, the single Switch Control Unit (SCU) routine, and the multiple SCU routine. These routines are illustrated in Figures 39, 40 and 41. The System Initialization routine utilizes two-fixed data tables and a set of variable data tables. The fixed-data tables include a directory of each trunk group in the system. It also has a directory that includes the system addresses of each switch control unit in the system. Each trunk group in the system also has a table; for example, trunk group number one would have a trunk group number one status table. Trunk group number two would be identified by a trunk group number two status table. In addition to having fixed data, the trunk group status tables also contain variable data. Variable data indicates whether the trunk is busy and if so what calling party is using the trunk. The purpose of the System Initialization routine is to clear out the variable data in the tables and to send to each switch control unit (SCU) a message to clear (disconnect) all crosspoints.

The single SCU routine requires access only to the SCU directory table in order to obtain the system address of the SCU. The multiple SCU routine requires access to the SCU directory table and to a table called "Inter-Matrix Path Route Table" that lists the

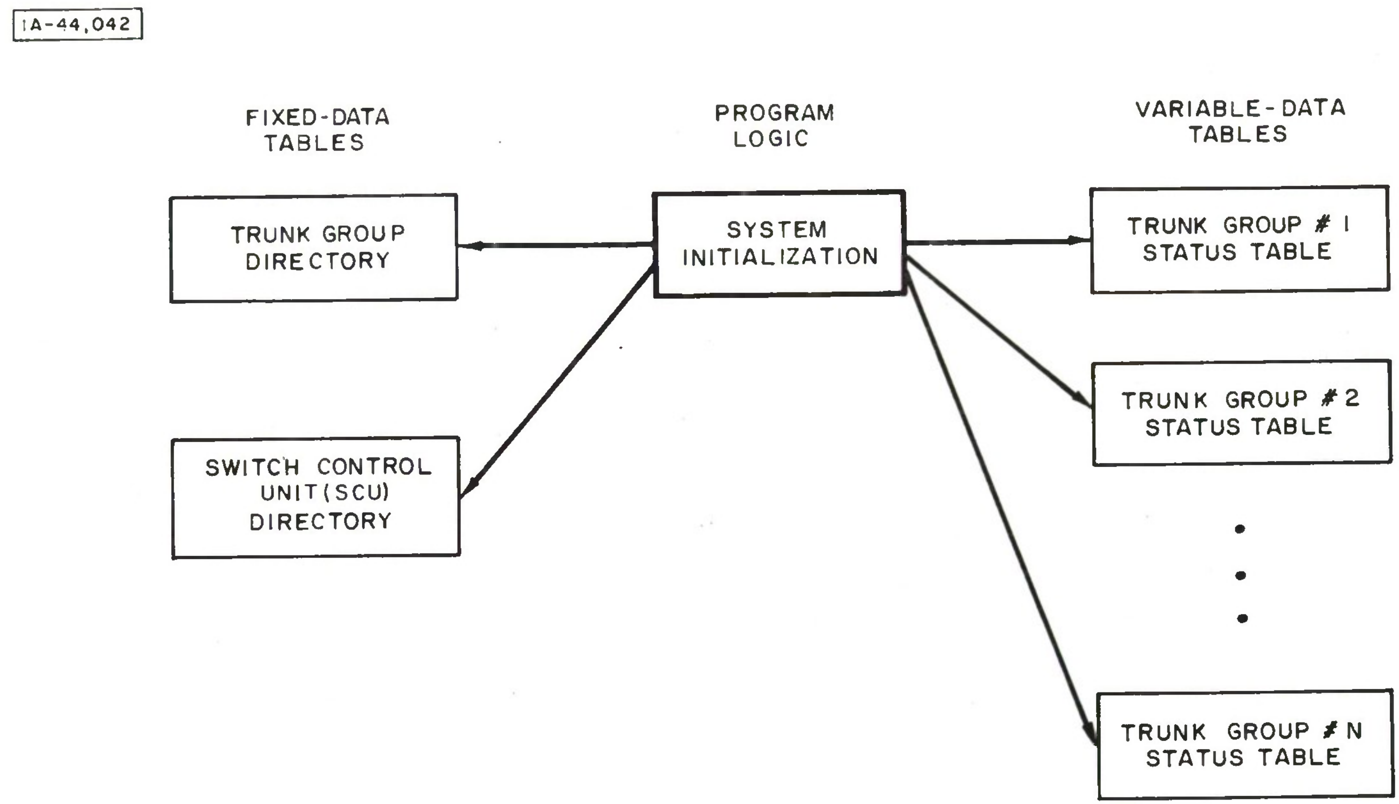


Figure 39 PATH System Initialization Routine

IA-44,043

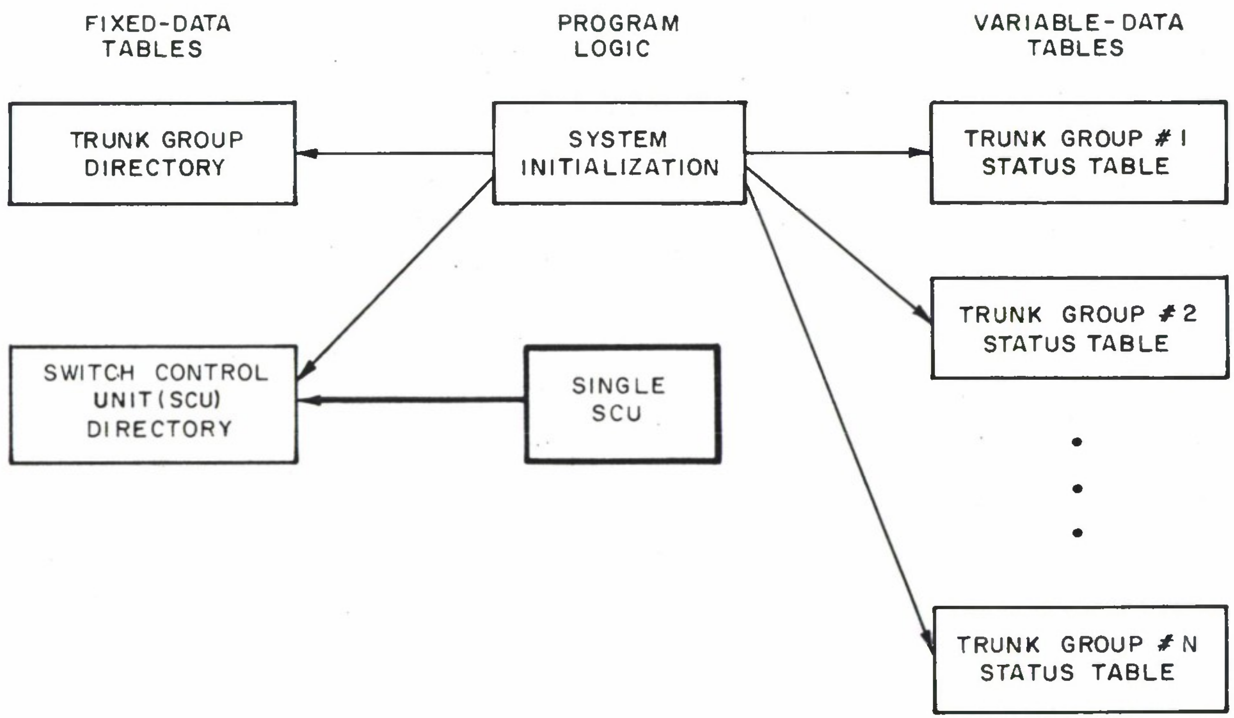


Figure 40 PATH Single Switch Control Unit Routine

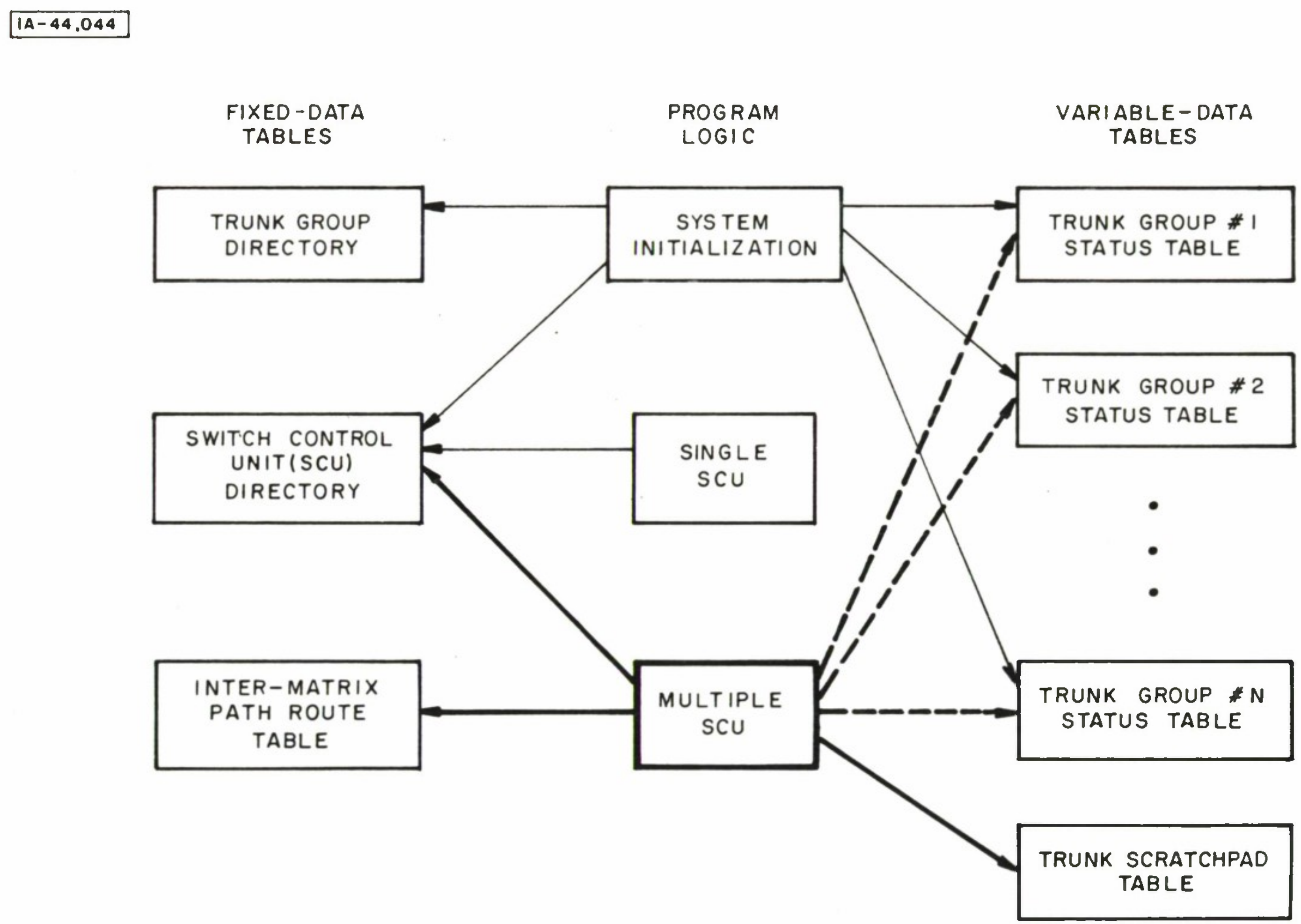


Figure 41 PATH Multiple Switch Control Unit Routine

available paths between any pair of matrices in the system. To determine the available trunks between matrices, this routine accesses one or more of the trunk group status tables. It also uses a scratch pad table to store the trunk(s) found during the search of the trunk group status tables. It then later retrieves these trunk identities from the scratch pad table when it is formulating its messages to the switch control units.

4.3.5.3 Program Logic. The program logic used in the Path Selection subroutine is briefly described below. Additional details on the program logic are shown in a condensed version of the programmer's detailed flowcharts in Appendix D.

The program logic checks to see if a "clear all crosspoints" command has been issued; if so, it interprets this as an initialization command. It then determines if a specific switch control unit has been identified, as might be the case in a special diagnostics program. If the switch control unit word is all zeros, it interprets this as a system initialization command and it formats the SCU message as "clear-all-crosspoints." It then goes to the SCU directory and obtains the first system address. It calls the subroutine SASO and gives it the first system address and the "clear-all-crosspoints" message. PATH then checks to see whether the last message has been sent out. If it has not, PATH goes to the SCU directory again and obtains the next system address and sends out the next message, and so on until all the switch control unit messages have been sent. When the SCU messages are completed, PATH goes to the trunk group directory, obtains the address of the first trunk group status table, and clears all of the variable information in that table. It proceeds to each trunk group status table in turn, until all the variable data in each table is cleared. PATH then returns control back to the Keypad Evaluation routine via the RTS software instruction.

If the control word does not contain a "clear-all-crosspoints" command, the program logic of PATH examines the subroutine call from KEYPAD to determine whether the switching matrix of the calling party and the called party are the same. If they are the same, it interprets the call to mean only one SCU message is required. PATH then determines the crosspoint to connect the calling and the called parties from their terminations, formats the message, and sends the single message via the SASO subroutine. As an example the connection of a video camera to a video monitor, both terminated on the same switching matrix, is illustrated in Figure 42. PATH then returns control back to KEYPAD via the RTS instruction.

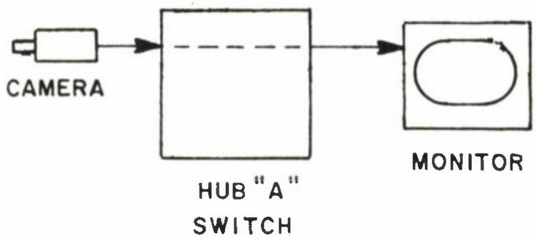


Figure 42A SINGLE SCU CONNECTION

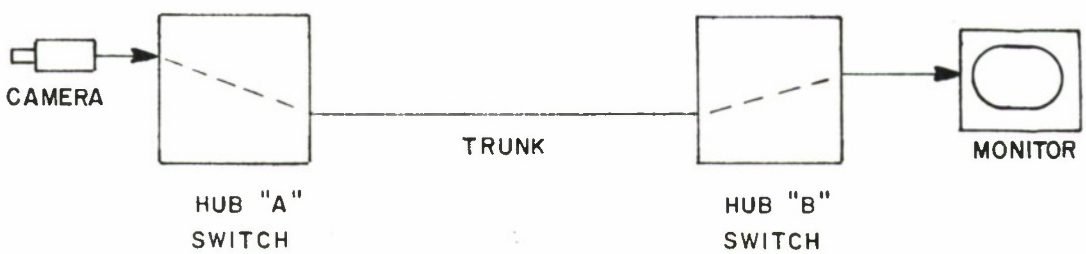


Figure 42B MULTIPLE SCU CONNECTION:
INITIAL SERVICE REQUEST

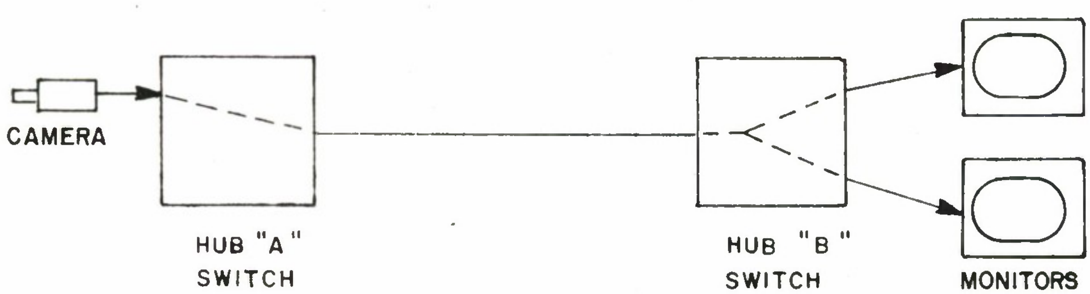


Figure 42C MULTIPLE SCU CONNECTION
"MODIFICATION: ADD" SERVICE REQUEST

IA—44,417

If the PATH program logic determines that the terminations are not on the same matrix, it then goes to the multiple SCU routine. In the multiple SCU routine, the program logic searches an inter-matrix path routing table for a match between the calling party and the called party matrices listed in a particular route. Since all routes are contained in the table, a match will be found. The found route in the path table indicates how many trunk groups are included in the path. This trunk group count is then moved to the scratch pad table for future use. The PATH program logic then looks at the command word to see if a connect function or a disconnect function is involved. The connect function is discussed first. In the connect function, the program logic goes to the found route in the inter-matrix path route table (IPRT) and obtains the pointer to the first trunk group in the path. It then searches the corresponding trunk group status table to determine if the calling party is already connected to one of the trunks.

If, for example, a video camera on Hub "A" is to be connected to a video monitor on Hub "B", the IPRT table would indicate which trunk group interconnects these two hubs and a connection would be required as indicated in Figure 42. As part of the software process, PATH will store in its trunk group status table the information that the calling party (camera) termination is transmitting on a particular trunk. Later, if another video monitor is to be connected to the same camera and the original connection is still to be maintained, the same trunk would be used. In other words, PATH would cause a multiple connection that is a parallel connection to be made by the switch control unit at Hub "B," as illustrated in Figure 42.

If in searching a Trunk Group Status Table (TGST) listed in the intermatrix path route table (IPRT), PATH finds that one of the trunks is already in use by the calling party, it interprets this as a request by the user for a Call Modification "Add" service request. If no cross-reference is found, the trunk group table is searched for the first idle trunk. When the first idle trunk is found, its address is stored in the scratch pad table. If no idle trunk is found, the search is discontinued and a busy indication is sent back to KEYPAD; i.e, the connection cannot be completed because all of the trunks in that path are busy.

Assuming the trunk has been found and stored in the scratch pad table, the PATH program logic then checks to see if that was the last trunk group status table to be searched; if not, it returns to the inter-matrix path route table, obtains the next trunk status group table and starts a search of that table. If it was the last

trunk group status table, then the PATH program jumps to a subsidiary subroutine of PATH called Set-Up-Switch-Messages (SUSM). This subroutine is used only by the PATH multiple SCU routine.

Up to this point, PATH knows the calling party termination, called party termination, and the intermediate trunks to be used in interconnecting the SCUs. The SUSM subroutine uses this information to formulate a message to each SCU in the path. In formatting and sending out the messages, its starts with the last SCU in the path because the called party may be sharing one or more of the trunks with other called parties. This connectivity situation is typical of a "broadcast" connection in which the calling party is simultaneously connected to two or more called parties. In setting up the message, therefore, the status of the trunk is determined as to whether it is an idle trunk or a trunk already in use by the calling party. If it is an idle trunk, the trunk end termination and the called party termination are connected via the last SCU and the next SCU in tandem is sent a message to connect the proper crosspoints. If, on the other hand, the trunk is not idle, (that is, it is marked as "in use" by the calling party), only the last SCU is sent a message. There is no need to connect any more matrices because they have already been connected under the prior "broadcast" connection.

In summary, SUSM checks the last SCU, sends out that message, and then looks at the trunk user count to determine if the next SCU requires a connection. If so, it sends out that message and then looks at the next trunk's user count to see if it is in use or not. It sends out messages to tandem SCUs until it reaches the first SCU in the path or a trunk that has already been connected to the calling party on a previous "broadcast" connection. At that point, the SUSM subroutine has completed its execution and it returns control back to the PATH "connect" routine sequence. The PATH "connect" routine sequence still has to update the variable data portion on the trunks used in the end-to-end path. It accomplishes this by going to the trunk scratch pad table, obtaining the location of the first trunk from the table, and incrementing the user count for that trunk in the trunk groups status table. It also adds the calling party termination to the cross-reference slot in the trunk group status table. It does this for each trunk identified by the trunk scratch pad table. When this task is completed, the connect sequence is finished and PATH returns control back to KEYPAD.

The PATH Disconnect routine starts out by searching the appropriate trunk group status table for a cross- reference termination match. If no match is found, it branches to a

diagnostic routine. Assuming a match is found, PATH stores the trunk identity in the scratch pad table. It then decrements the trunk user count. If the decrement produces a zero-use count, the program logic clears the calling party cross-reference termination from that slot in the trunk group status table. The PATH disconnect routine then checks to see if all the applicable trunk group status tables have been searched. If not, it proceeds to the next trunk groups status table identified in the inter-matrix path table (IPRT).

When all the trunks have been found, the SUSM subroutine is again called. After the SUSM subroutine has sent all the switch control unit disconnect messages, it then returns to the PATH disconnect routine. PATH immediately returns control to the Keypad Evaluation routine.

The Disconnect routine is somewhat different from the Connect routine in the sense that Disconnect updates the status tables before it calls the SUSM subroutine. This is a convenience in the program logic and does not have any system connotations. It permits the exact subroutine logic coded in SUSM to be used for the both the connect messages and the disconnect messages, thus saving program coding and program debugging.

4.3.6 Dispatch/Scheduler

The NETCOP program routines can obtain control of the processor in one of three ways: (1) by an interrupt, (2) directly from another major routine, or (3) from DISPATCH.

Routines that are run as the result of an interrupt are high priority routines that must be executed within a short period of time. The primary interrupt-driven routine in NETCOP is the signaling and supervision input (SASI) that accepts and stores keypad digits from the hardware. Upon completion of an interrupt-driven routine, control is passed back to the routine that was interrupted. A routine can be given control directly from another major routine; after its job is done, it may or may not transfer control back to the first routine. This is the normal programming procedure.

DISPATCH and SCHEDULER are used to control the execution of low priority routines. The Scheduler is the subroutine called when one routine wants work done by another routine, but doesn't want to wait for the other routine to execute. The Scheduler is given the name of the routine to be run and the location of its input data; it

records the fact that the routine is to be run, saves the input data in the storage pool, and then transfers control back to the calling routine without executing the scheduled routine.

Execution of scheduled routines is controlled by DISPATCH, which is the routine that runs continuously even when there is nothing else to be done. Its input is a list of routines which can be scheduled. It searches this list for the highest priority routine to be run, and transfers control to that routine. After the routine is executed, control is transferred back to DISPATCH which destroys the input data for the routine just executed and searches for another scheduled routine.

The most important routine under control of the SCHEDULER and DISPATCH is Keypad Evaluation. It is scheduled by the SASI routine at the completion of a calling sequence. The only other scheduled routine at present is LAMP, a routine that controls indicator lights on the keypad. LAMP is scheduled by either SASI or KEYPAD.

Additional details on the program logic are shown in a condensed version of the programmer's detailed flowcharts in Appendix E.

4.3.7 Pool Handler

The "pool" is a large block of unassigned memory that is used to contain all variable length tables in the NETCOP program software. The pool is divided into "nodes" that are fixed length buffers containing two pointers and seven words for data storage (Figure 43). A node can contain data from any variable length table. A given node can be used by one routine for storage, can then be relinquished, and thereafter used by a different routine for different data. Nodes can also be used for passing data from one routine to another.

Each node in the pool forms a link in one of several logical chains. A logical chain contains all of the data in a variable length table. In two consecutive words in memory (outside of the pool), there are pointers to the nodes that form the first and last links of the chain. These are called "first pointer" and "last pointer." Each link contains a pointer to the preceding link and a pointer to the following link. There is no limit to the length of a chain. The absence of a chain is represented by "-1" in both first and last pointers.

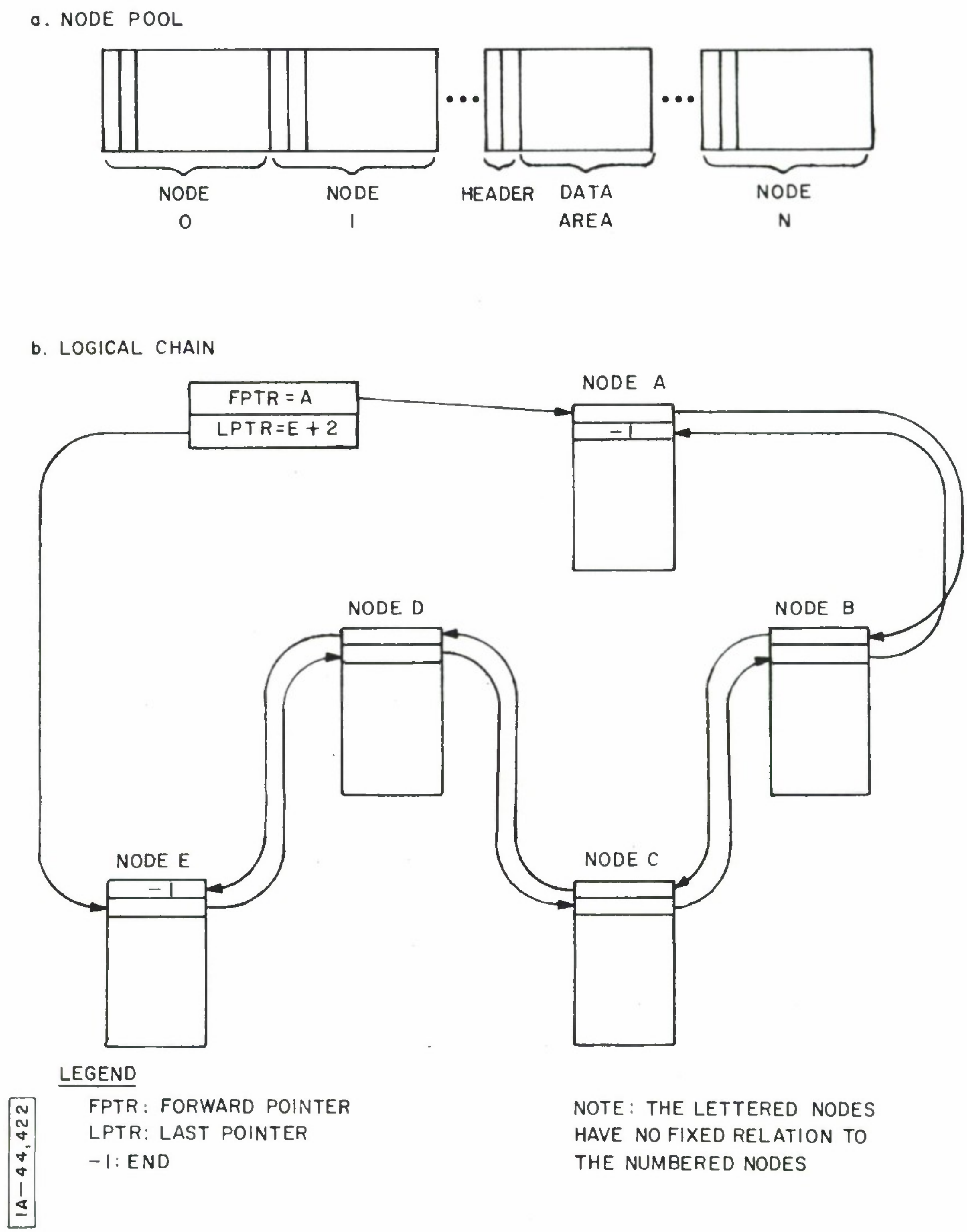


Figure 43 POOL CONSTRUCTION

At initialization, all nodes in the pool are linked together to form the one long chain that is assigned to the Pool Handler, the routine which manages the pool (Figure 43). These nodes are empty; the Pool Handler itself contains no variable length tables. When the system is in operation, nodes will be assigned to SASI, the Keypad Evaluation Routine, LAMP (indicator light routine), and to any user who has a call in progress. First and last pointers of chains belonging to a routine (SASI, KEYPAD, LAMP, POOL) are contained in a subroutine header that is assembled at the very beginning of the routine. First and last pointers of chains belonging to users are contained in the user data table. The short title for the Pool Handler routine is POOL.

The job of POOL is to transfer a link from one chain to another. If a node is needed for data storage, a call is made to POOL to transfer a node from POOL's own chain (containing empty nodes) to the calling routine's chain. A node can be relinquished by transferring it back to POOL. Transfer of information from one routine to another is accomplished by telling POOL the location of the node containing the information, the location of the pointers of the present owner of the chain, and the location of the pointers of the new owner. A node can be added to the beginning or end of a chain by telling POOL "add to beginning" or "add to end." A node can be deleted from the beginning or end of a chain in a similar fashion or a specific link in the middle of a chain can be deleted by telling POOL its address. POOL never actually moves data from one node to another, but accomplishes all transfers by changing pointers in the nodes and/or the routines that own the nodes (Figure 44).

Additional details on the program logic are shown in a condensed version of the programmer's detailed flowcharts in Appendix F.

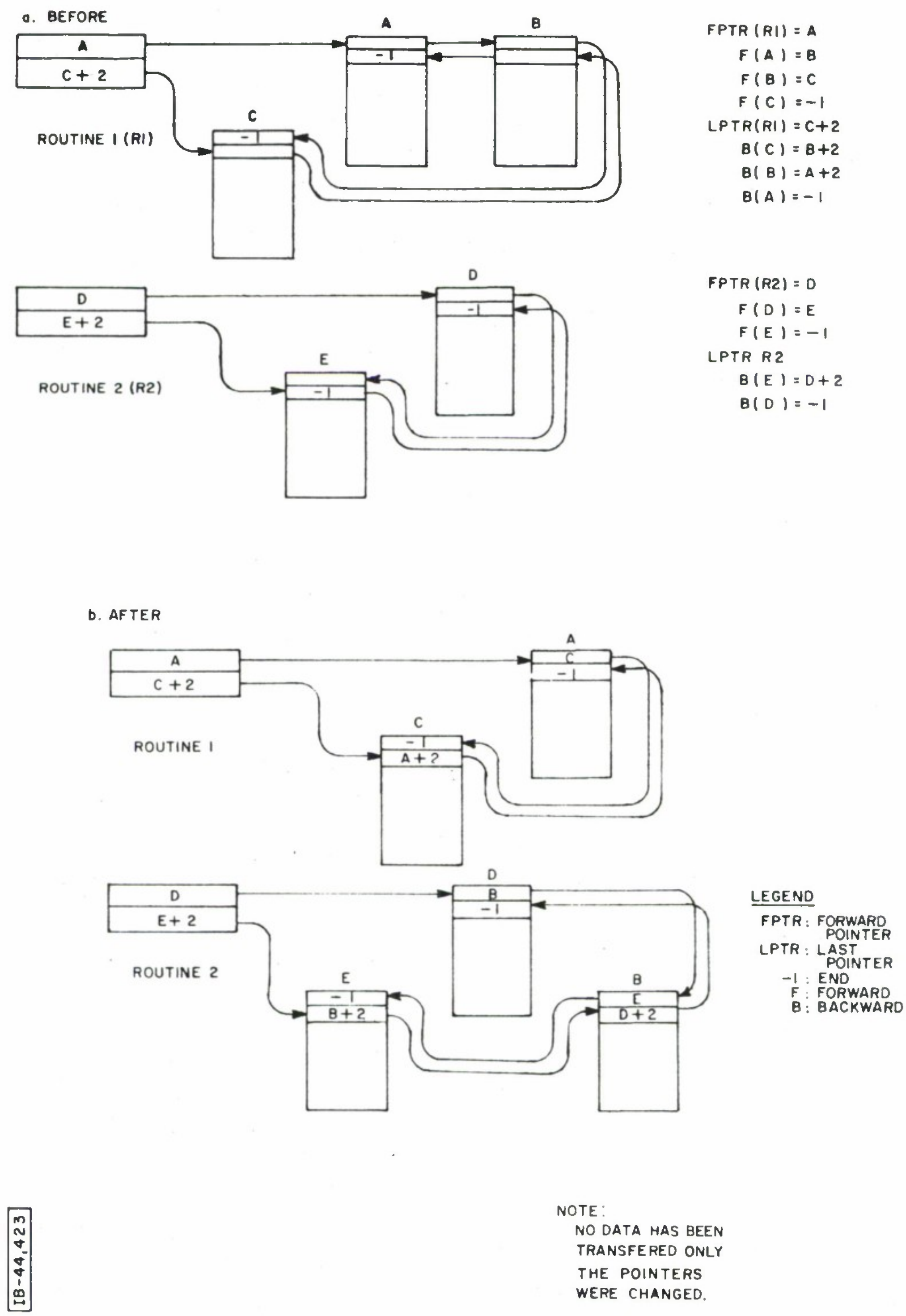


Figure 44 NODE MOVE (MOVE B TO AFTER D)

Section V

5.0 LABORATORY EVALUATION CONFIGURATION

5.1 Background

The AFBITS experimental laboratory activities are directed toward developing and evaluating design techniques for the hardware and software elements of a network central control system. The software task primarily entails the programming of a PDP-11 minicomputer, and the hardware task involves the design and construction of a signaling and supervision system that allows the user to send request-for-service information to the PDP-11 and receive back system status information. For purposes of saving time and money, a limited set of the software programs was coded and a limited number of keypad logic interface units were built.

5.2 Test Configurations

A number of computer and communications-related devices are available for use in the AFBITS multi-mode system demonstration laboratory (Reference 1). The devices selected allowed for the formulating, programming and coding of a variety of service packages.

The interconnection of the six work station test configurations presently used in the laboratory are illustrated in Figure 45. Three varieties of video monitors were available for evaluation. The first work station (WS #1) contains a high resolution video monitor operating at 1029 "TV" scan lines per frame. The second work station uses a standard resolution video monitor operating at 525 "TV" scan lines. The third work station uses a monitor that can be switched automatically between high resolution and standard resolution, depending on the video signal being received. A keyboard is also available for use with the monitors. In Figure 45 the keyboard is colocated with the monitor in Work Station #2. However, it can be moved to either Work Station #1 or #3 to suit any particular demonstration or experiment. The addition of the signaling and supervision keypads permits these devices to be used as visual display terminals for a variety of work station applications. For example, it is possible to demonstrate at a single terminal the automatic retrieval of microfiche, the remote control of a surveillance camera and the entering of alphanumeric data.

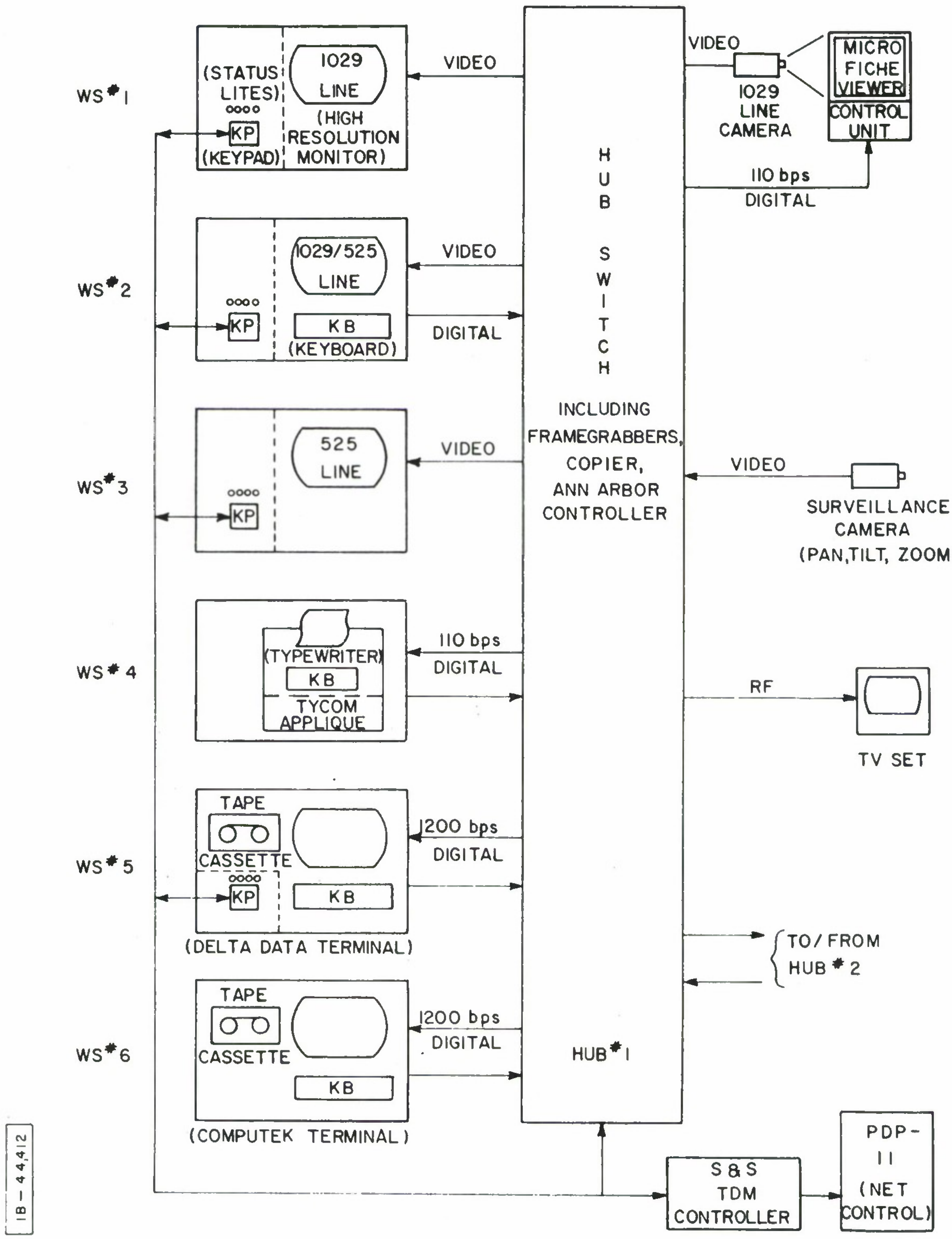


Figure 45 WORK STATIONS-LABORATORY EVALUATION TEST CONFIGURATION

The fourth work station configuration uses a standard IBM Selectric typewriter equipped with a Tycom, Inc. manufactured applique that permits sending and receiving typed information at 110 bit per second speed. The Selectric typewriter is shown mounted on the Tycom applique along with the Tycom power supply and the coaxial cable interface in Figure 46. Due to the limited availability of signaling and supervision keypads, the fourth work station was not supplied with a keypad. This prevents WS #4 from originating connections, but it is fully capable of receiving and transmitting data after another work station establishes a connection to it.

The fifth work station configuration uses a standard visual display terminal manufactured by Delta Data Inc. that is intended primarily for interfacing with a remote computer (host processor). The Delta Data terminal is equipped with a dual magnetic tape cassette unit and is capable of line speeds between 110 bps and 9600 bps, selected by its own manual switch control (Figure 47).

The sixth work station contains a smart (intelligent) terminal, that is a terminal with a built-in <u>programmable</u> microprocessor manufactured by Computek Inc. It is equipped with a dual tape cassette unit and operates at line speeds between 110 bps and 9600 bps, under its own manual switch selection control (Figure 48).

Two common-user video signal source devices are connected to the evaluation system. The first is a combination of a microfiche viewer and a high resolution camera with a built-in control unit capable of accepting ASCII code that permits control from a remote terminal (Figure 49). The microfiche viewer also has an associated hardcopy printing device. The second video device is a remotely controlled surveillance camera with pan, tilt and zoom features.

For simulating a tree type subscriber being accessed from a hub type subscriber, a TV set was also connected to the hub equipment. The TV set, with text inputted from its associated data entry and retrieval keyboard, is shown in Figure 50.

Internal to the hub are the common equipments that are part of the overall hub switch assembly. These common equipments include two types of video framegrabbers: a low cost, standard resolution (525 line) unit manufactured by the Hitachi Company and a more expensive, high resolution (1029 line) unit manufactured by Princeton Electronics Products, Inc. Associated with the high resolution unit is a video copier manufactured by Tektronix, Inc. The framegrabber/copier combination permits hardcopy to be physically

Figure 46 IBM Selectric Typewriter with TYCOM Applique

Figure 47 Delta Data Terminal with Magnetic Tape Cassette

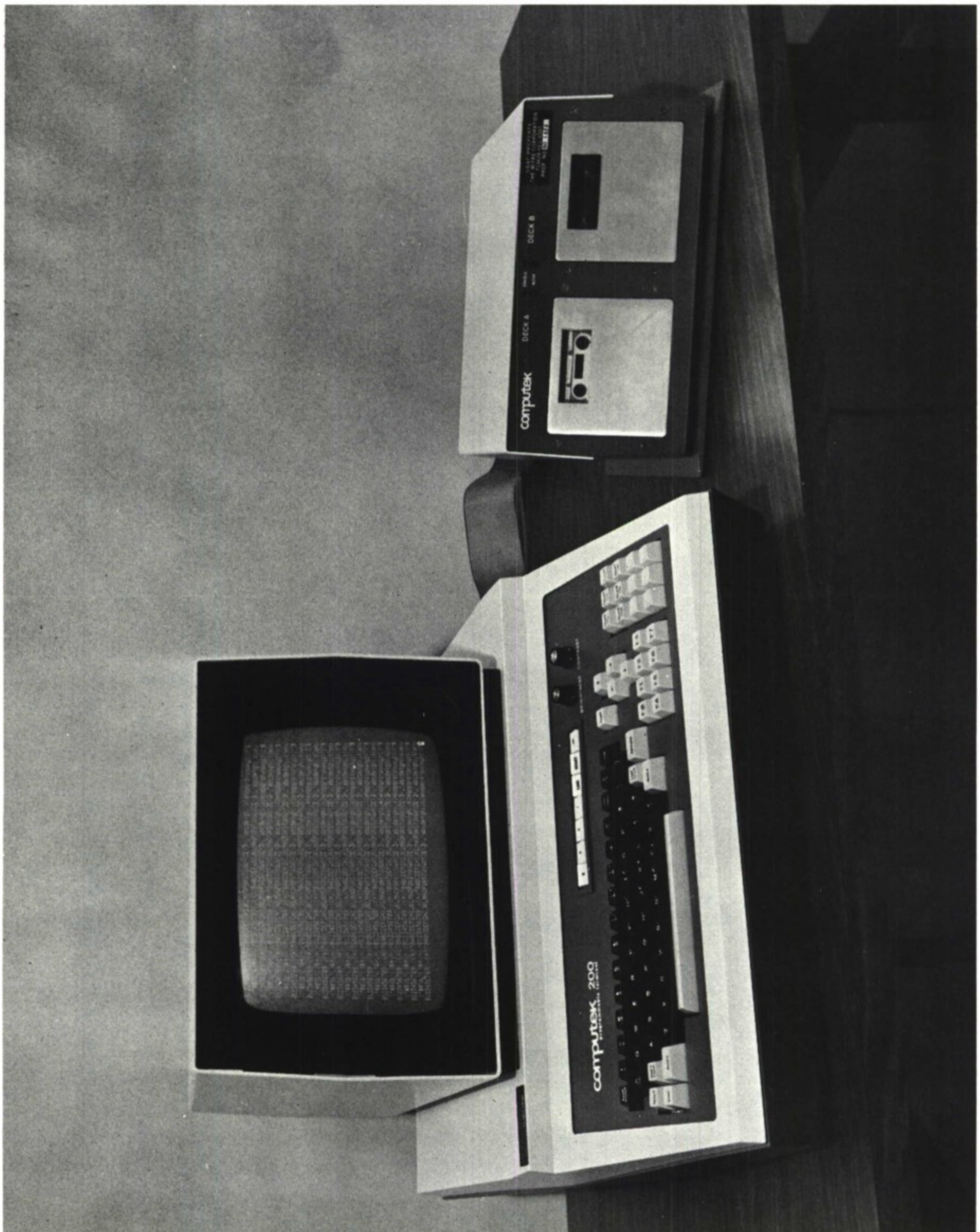


Figure 48 COMPUTEK Terminal with Magnetic Tape Cassette

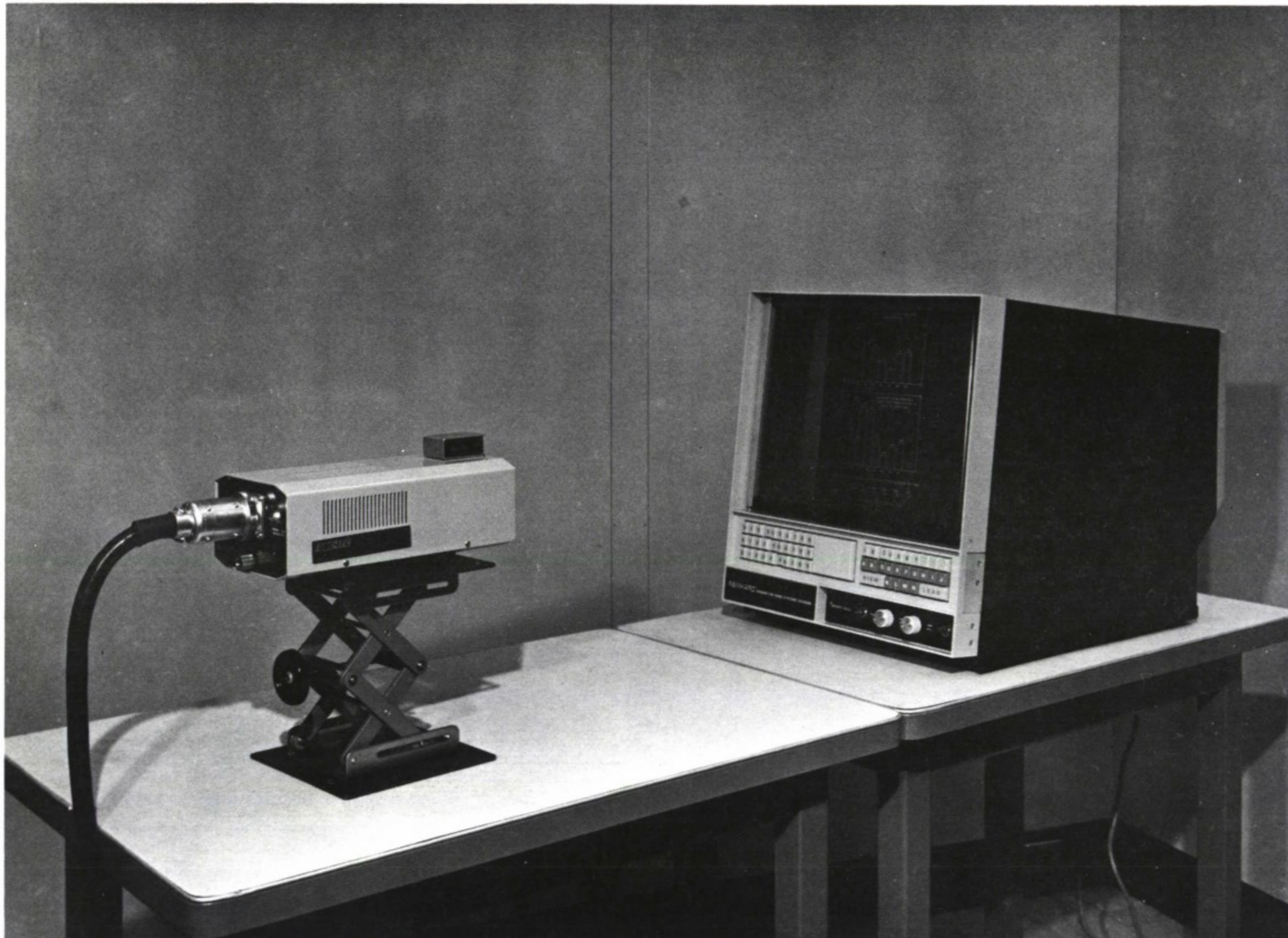


Figure 49 Microfiche Viewer and High Resolution Camera

Figure 50 Remote TV Set Terminal and Keyboard

closer to the user, as compared to obtaining copies from a remote microfiche machine.

Another hub common-user equipment is a combination "character generator-refresh memory" controller that accepts digital data and generates, simultaneously, a baseband video signal and a TV (rf-modulated) video signal. The latter signal is for connection to the previously mentioned "tree" TV terminal. The controller is manufactured by Ann Arbor Inc. The heart of the hub switch is a 20 X 20 reed relay, video switching matrix manufactured by Matrix Systems, Inc. that features a bandwidth in excess of 60 megahertz. The matrix has as its switch control unit a passive logic unit with the same type of interface to the coaxial cable that the user has at his keypad. The switch control unit interface is in effect a "keyless keypad" interface. A second hub, with a smaller 10 x 10 video matrix interconnects with this hub. Since the software was designed and programmed for a multi-hub system, the combination of these two hubs allows for complete exercise of the software.

The AFBITS laboratory evaluation configuration network controller, Figure 51, uses a PDP-11 minicomputer manufactured by Digital Equipment Corporation (DEC). The PDP-11 is a 16-bit, general purpose computer representative of modern minicomputers. Communication between computer elements and the AFBITS signaling and supervision controller is via the input/output high-speed bus that DEC calls the UNIBUS. The minicomputer architecture is designed around the bus. The processor, memory and all peripheral devices share the same bus. This permits all devices to be addressed as if they were main memory, thus greatly simplifying the programming of input/output operations. Further enhancing the ease of input/output operations is the Interrupt structure via the bus that permits the processor to be interrupted by a peripheral device. For example if a printer is ready for another character, it can signal an Interrupt and the processor can automatically "vector" to the printer handler program. However, the operating program and the interrupt program have their own priority levels. If the portion of the operating program to be interrupted has a higher priority level than the Interrupt, the processor will ignore the Interrupt until it reaches a lower priority portion of the operating program. (The alternative to the Interrupt approach is called Direct Input/Output. Direct I/O requires continuous checking of a peripheral by the processor to determine if the peripheral is ready for the next character or word transfer. For most I/O operations the use of the Interrupt feature is much more efficient than using Direct I/O.)

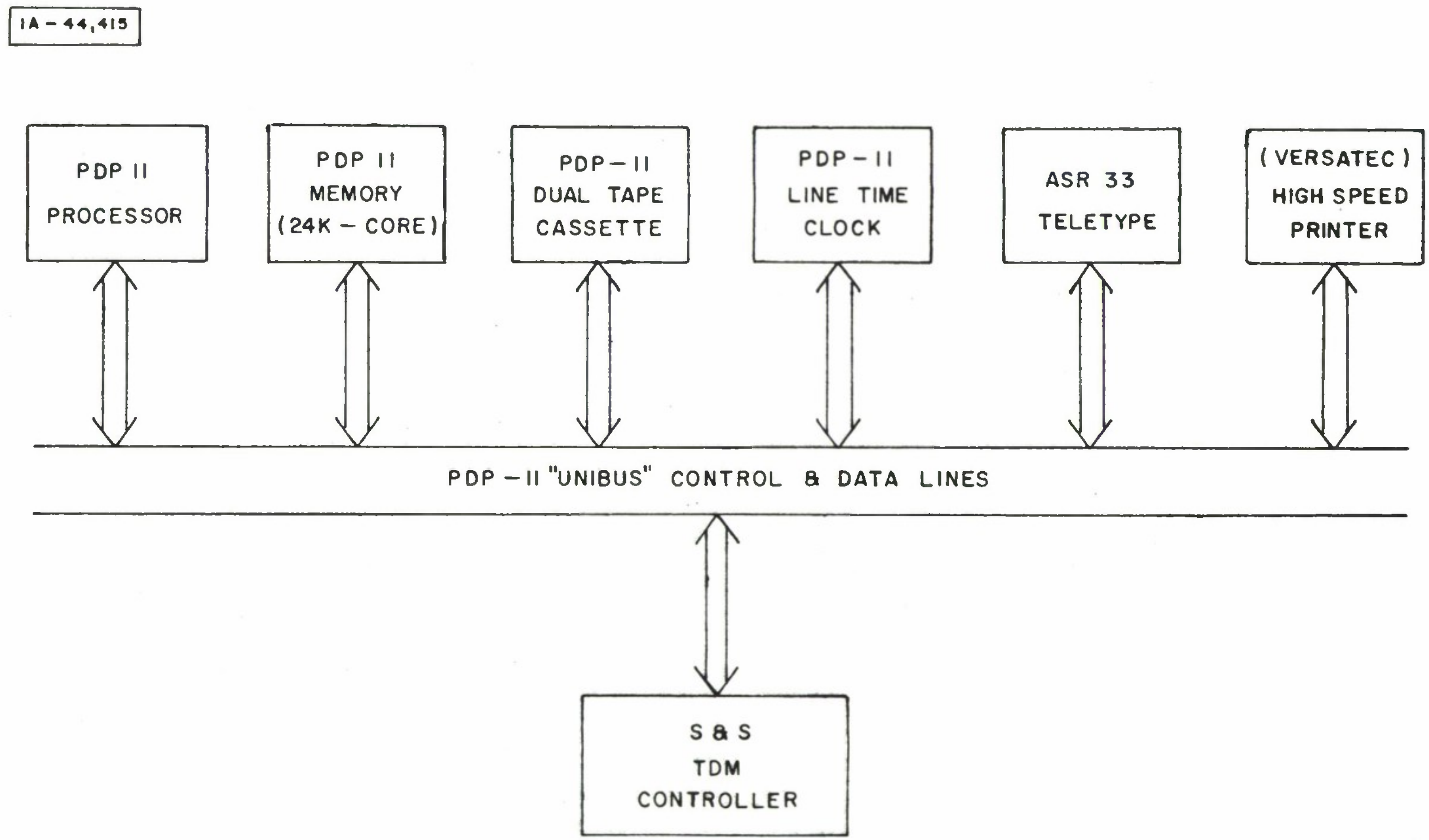


Figure 51 AFBITS LABORATORY MINICOMPUTER CONFIGURATION

Other attractive features of the PDP-11 include the provision of eight general purpose registers within the central processor that can be used as accumulators, index registers or address pointers and the ability to code two addresses within a single instruction. The overall result of these advantageous features is increased system throughput and decreased programming time.

The equipments briefly described above are representative of a wide variety of equipments available in the data and video distribution fields. The laboratory activity has been directed toward demonstrating the technical feasibility of interconnecting equipment types rather than the specific equipments themselves. From the wide variety of service combinations available, eight specific sequences were chosen for laboratory demonstration and evaluation. These included the following:

1. Surveillance camera to originator's (calling party's) monitor.

2. Surveillance camera to calling party's monitor plus a parallel connection to the input of a Hitachi framegrabber with its output connected to a destination (called party's) monitor.

3. A microfiche viewer connection to a user's terminal via an intermediate framegrabber connection.

4. A microfiche viewer connection to a user's terminal without an intermediate framegrabber.

5. Digital data entry via the user's keyboard to the Ann Arbor character-generator controller with the video output connected to the user's monitor.

6. A low-speed digital data full duplex connection.

7. Low speed data half-duplex connection with the calling party's low-speed output being transmitted to the called party's low-speed input.

8. Low-speed digital data half-duplex connection with the called party's low-speed output connected to the calling party's low-speed input.

This selection of service combinations permits testing of basic service packages as well as alternate service packages. For example, the alternate service package for the full-duplex low-speed

connection of number six is the half-duplex connection of number seven. This alternate service package might be utilized if a full-duplex connection was not available. The next section presents additional details of the specific service package connections used in the laboratory demonstrations.

5.3 Service Packages Used In Laboratory Experimental Demonstrations

A specific dialing sequence is used for each service package. Where there is only one end item accessible to the requesting user, the last four digits are 0000. However, when there is a variety of accessible end items, the last four digits represent the directory address of the end items. If an alternative service package is available, it is so indicated. The processor action required to effect the service package is also indicated. Finally, the modifications permissible under the service package are identified. These basically include adding parties to a connection or dropping parties that may have been added previously. The eight service package options are specified below.

1. Surveillance Camera to Calling party's monitor.
    - (a) Dialing Sequence: S 01 0000
    - (b) Alternate Service Package: None.
    - (c) Processor Action: Surveillance camera's video output termination is connected to calling party's video monitor.
    - (d) Modifications:
        1. Add another party to video connection.
            - (a) Dialing Sequence: M 1 S 01 XXXX
            - (b) Processor Action: Surveillance camera's video output termination is connected to called party's monitor.
        2. Drop party that was added by call modification.
            - (a) Dialing Sequence: M 2 S 01 XXXX
            - (b) Processor Action: Surveillance camera's video output termination is disconnected from called party's monitor.

2. Surveillance camera to calling party´s monitor through Hitachi framegrabber to called party´s monitor.
    (a) Dialing Sequence: S 02 XXXX
    (b) Alternate Service Package: None.
    (c) Processor Action:
        1. Surveillance camera´s video output termination is connected to calling party´s monitor.
        2. Surveillance camera´s video output termination is connected to Hitachi framegrabber´s video input termination.
        3. Hitachi framegrabber´s video output termination is connected to called party´s monitor.

    (d) Modifications:
        1. Add another party to video connection.
            (a) Dialing Sequence: M 1 S 02 XXXX
            (b) Processor Action: Hitachi framegrabber´s video output termination is connected to called party´s monitor.
        2. Drop party that was added by call modification.
            (a) Dialing Sequence: M 2 S 02 XXXX
            (b) Processor Action: Framegrabber´s video output termination is disconnected from called party´s monitor.

3. Microfiche with framegrabber.
    (a) Dialing Sequence: S 03 0000
    (b) Alternate Service Package: #4
    (c) Processor Action:
        1. Calling party´s keyboard is connected to the microfiche´s low-speed digital data input termination.
        2. Microfiche´s video output termination is connected to the framegrabber´s video input termination.
        3. The framegrabber´s video output termination is connected to the calling party´s video monitor.
    (d) Modifications:
        1. Add another party to video connection.
            a. Dialing Sequence: M 1 S 03 XXXX
            b. Processor Action: Framegrabber´s video output termination is connected to called party´s monitor.
        2. Drop party that was added by call modification.
            a. Dialing Sequence: M 2 S 03 XXXX
            b. Processor Action: Framegrabber´s video output termination is disconnected from called party´s monitor.

4. Microfiche without framegrabber.
    (a) Dialing Sequence: S 04 0000
    (b) Alternate Service Package: None.

(c) Processor Action:
   1. Calling party's keyboard is connected to the microfiche's low-speed digital data input termination.
   2. The microfiche's video output termination is connected to calling party's monitor.

(d) Modifications:
   1. Add another party to video connection.
      (a) Dialing Sequence: M 1 S 04 XXXX
      (b) Processor Action: The microfiche's video output termination is connected to the called party's monitor.
   2. Drop party that was added by call modification.
      (a) Dialing Sequence: M 2 S 04 XXXX
      (b) Processor Action: The microfiche's video output termination is disconnected from the called party's monitor.

5. Digital Data entry: from user's keyboard.
   (a) Dialing Sequence: S 05 0000
   (b) Alternate Service Package: none
   (c) Processor Action:
      1. Calling party's keyboard is connected to the Ann Arbor character-generator controller's low speed input termination.
      2. Ann Arbor controller's video output termination is connected to calling party's video monitor.
   (d) Modifications:
      1. Add another party to video connection.
         (a) Dialing Sequence: M 1 S 05 XXXX
         (b) Processor Action: Ann Arbor Controller's video output termination is connected to called party's video monitor.
      2. Drop party that was added by call modification.
         (a) Dialing Sequence: M 2 S 05 XXXX
         (b) Processor Action: Ann Arbor Controller's video output termination is disconnected from called party's video monitor.

6. Low-speed digital data: full-duplex connection.
   (a) Dialing Sequence: S 06 XXXX
   (b) Alternate Service Package: #7
   (c) Processor Action:
      1. Calling party's low-speed digital data output termination is connected to called party's low-speed digital data input termination.
      2. Called party's low-speed digital data output termination is connected to calling party's low-speed digital data input termination.

(d) Modifications:
   1. Add another party to low-speed digital data output connection.
      (a) Dialing Sequence: M 1 S 06 XXXX
      (b) Processor Action: Calling party's low-speed digital data output termination is connected to called party's low-speed digital data input termination.
   2. Drop party that was added by call modification.
      (a) Dialing Sequence: M 2 S 06 XXXX
      (b) Processor Action: Calling party's low-speed digital data output termination is disconnected from called party's low-speed digital data input termination.

7. Low-speed digital data: calling party's low-speed digital data output to called party's low-speed digital data input.
   (a) Dialing Sequence: S 07 XXXX
   (b) Alternate Service Package: None.
   (c) Processor Action: Calling party's low-speed digital data output termination is connected to called party's low-speed digital data input termination.
   (d) Modifications:
      1. Add another party to low-speed digital data output connection.
         (a) Dialing Sequence: M 1 S 07 XXXX
         (b) Processor Action: Calling party's low-speed digital data output termination is connected to called party's low-speed digital data input termination.
      2. Drop party that was added by call modification.
         (a) Dialing Sequence: M2S07 XXXX
         (b) Processor Action: Calling party's low-speed output termination is disconnected from called party's low-speed digital data input termination.

8. Low-speed digital data: called party's low-speed digital data output to calling party's low-speed digital data input.
   (a) Dialing Sequence: S 08 XXXX
   (b) Alternate Service Package: None.
   (c) Processor Action: Called party's low-speed digital data output termination is connected to calling party's low-speed digital data termination.
   (d) Modifications: None.

The equipments used in the experimental evaluations are summarized in Tables I and II. The common equipments and their associated switching matrix terminations are listed in Table I. The "work station" terminal equipments and their switching matrix terminations are listed in Table II.

Table I

Common Equipment Used in AFBITS Lab Demonstration

Equipment Number	Description	Unit #	User Catagories	Terminal Type	Matrix	Termination
1	Microfiche	1	1, 2	Low Speed Input	1	Y3
				Video Output	1	X12
2	Framegrabber (PEP)	1	1, 2	Video Input	1	Y11
				Video Output	1	X17
3	Surveillance Camera	1	1, 2	Video Output	1	X11
4	Hitachi Frame-grabber	1	1, 2	Video Input	1	Y5
				Video Output	1	X14
5	Ann Arbor Controller	1	1, 2	Low Speed Input	1	Y0
				Low Speed Output	1	X0
				Video Output	1	X10

Table II

Work Stations Used in AFBITS Lab Demonstration

User #	Description	User Catagory	Svc Pkgs Allowed	Terminal Type	Matrix	Termination
1	Keypad, Keyboard and 1029 Monitor	1	3,4,6,7	Video Input Low Speed Output	1 1	Y9 X5
2	Keypad, Keyboard and 1029/525 Monitor	1	1,3,4,5,6,7	Video Input Low Speed Output	1 1	Y8 X6
3	Keypad, Keyboard and 525 Monitor	1	1,2,5,6,7	Video Input Low Speed Output	1 1	Y7 X7
4	Typewriter (No Keypad)	1	None	Low Speed Input Low Speed Output	1 1	X4 Y4
5	Keypad and Delta Data Terminal	1	6,7,8	Low Speed Input Low Speed Output	1 1	X2 Y2

Table II

Work Stations Used in AFBITS Lab Demonstration
(Continued)

User #	Description	User Catagory	Svc Pkgs Allowed	Terminal Type	Matrix	Termination
6	Computek Terminal (No Keypad)	1	None	Low Speed Input Low Speed Output	1 1	X1 Y1
7	TV Monitor (No Keypad)	1	None	Video Input	1	Y10

APPENDIX A

SPECIFICATION OF HEADEND INTERFACE CONTROLLER REGISTERS

This appendix defines the Control and Status Words used in the AFBITS headend control hardware interface that links the Network Control Processor (NCP) to the downstream and upstream cable elements.

1. Downstream Status and Control Register (DSCR).

Address: 164104 (Octal)

Bit 0 (R/W can be read out or written into), GO. When set to "0," all down and upstream control action stops. When set to "1" all control signals, data, timing, polling, etc., commences.

Bit 1 (R/W), INIT. This bit provides for hardware resetting. When the transition from "0 to 1" takes place, a hardware RESET is generated to initialize and clear all buffer, FIFO´s, counters, etc.

This bit must be set to a "1" for the Computer START RESET (BUS INIT L) to cause clearing. The resulting reset action is identical to that described above. If the bit is held at "0," the BUS INIT L cannot reset the circuits.

The following truth table applies to the combination of GO (Bit 0) and INIT (Bit 1) being written into DSCR.

Bit 0	Bit 1	Result:
1	1	Delay GO until INIT occurs, then GO.
0	1	INIT only and continue waiting for GO.
1	0	Go immediately.
0	0	Stop going and freeze everything in place.

Bits 2 and 3 - (R/W). Software poll rate control bits. These bits control the frequency at which software polling can be interspersed with hardware polls.

Bit 2	Bit 3	Result:
1	1	Insert a software poll every 16th hardware poll.
0	1	Insert a software poll every 8th hardware poll.
1	0	Insert a software poll every 4th hardware poll.
0	0	This is referred to as a default case and is discussed below.

Default Case: When Bits 2 and 3 are both 0, the rate at which software polling will go on is determined and controlled dynamically by the number of software polls queued up in the FIFO. The algorithmic relationship that defines this is as follows:

# of polls in queue	Rate of interspersing
1 thru 9	every 16th hardware poll
10 thru 19	every 8th hardware poll
20 thru 29	every 4th hardware poll
30 thru 39	every 2nd hardware poll

Bits 4 and 5, not used.

Bit 6 (R/W) Interrupt Enable bit for Bit 7 - FIFO FULL.

Bit 7 - (Read Only) FIFO FULL.

Bit 8 - (Read Only) Buffer Word Count (B.W.C.). This bit is set to a "1" when the first word of a two word downstream message or software polling message is read into the downstream FIFO.

Bits 9 and 10, not used.

Bits 11, 12 and 13 (Read Only). Transmission Error Status bits.

If an error is detected in the downstream transmission and reported back to the headend, it is identified as follows:

Bit 11, Overrun Error.

Bit 12, Framing Error.

Bit 13, Parity Error.

Bit 14 (R/W) an Interrupt Enable bit for Bit 15.

Bit 15 is an ERROR OCCURRED bit which gets set on the occasion of any error.

2. Upstream Status and Control Word (USCR).

Address: 164100 (Octal).

Bit 0 - (R/W) Interrupt Enable for Bit 1.

Bit 1 - (Read Only), FIFO 1/4 FULL. This bit indicates that the upstream FIFO is 1/4 full. It is active for the range 1/4 to 1/2 and for 3/4 to full.

Bit 2 - (R/W), Interrupt Enable for Bit 3.

Bit 3 - (Read Only), FIFO 1/2 FULL. This bit indicates that upstream FIFO is between 1/2 full and completely full.

Bit 4 - (R/W) Interrupt Enable for Bit 5.

Bit 5 - (Read Only) FIFO FULL. This bit indicates that the upstream FIFO is completely full.

Bit 6 - (R/W) Interrupt Enable for Bit 7.

Bit 7 - (Read Only), FIFO READY. This bit indicates that the upstream FIFO is ready to be read, i.e. it contains at least one message. An additional characteristic is that a one millisecond delay will occur after a FIFO READY interrupt and before another FIFO READY interrupt can happen.

Bit 8 - (Read Only) Buffer Word Count (B.W.C.). This bit is set to "1" when the first word of a two word message is read out of the upstream FIFO.

Bit 9 - not used.

Bit 10 - (Read Only) Check Address Error bit. Check address refers to the 8 bit check address that is shipped back to the headend controller from the responding keypad. The bit sets when the error occurs and clears when the USCR is read out.

Bit 11, 12, and 13 (Read Only) respectively are Overun Error, Framing Error, and Parity Error. These three error conditions relate to the upstream transmission and are set on occurrence. They are reset when the Status Register is read out.

Bit 14 - (R/W) Interrupt Enable bit for Bit 15.

Bit 15 - (Read Only) is an Error flag which sets when any of the four error conditions (Bits 10 thru 13) are set.

3. Downstream Data Register (DDR).

Address: 164106 (Octal)

This register is addressed to receive each of the two 16 bit words which make up a software poll message.

4. Upstream Data Register (UDR).

Address: 164102 (Octal)

This register will contain a 16 bit word two of which make up the upstream message. When it is addressed for read out, the next 16 bit piece of an upstream message, either the second half of one message or the first half of the next message, is available.

APPENDIX B

SIGNALING AND SUPERVISION INPUT FLOWCHARTS

NOTE:

In accordance with conventional flowcharting techniques, special connector symbols are used to represent an entry from, or an exit to, another part of the program flowchart.

The symbol is used to indicate an exit from a page. The letter within the symbol indicates the entry point that will be found on another page.

The symbol ◯ is used to indicate an exit to another part of the program on the same page. It is also used in the following flowcharts to indicate the entry point from another page.

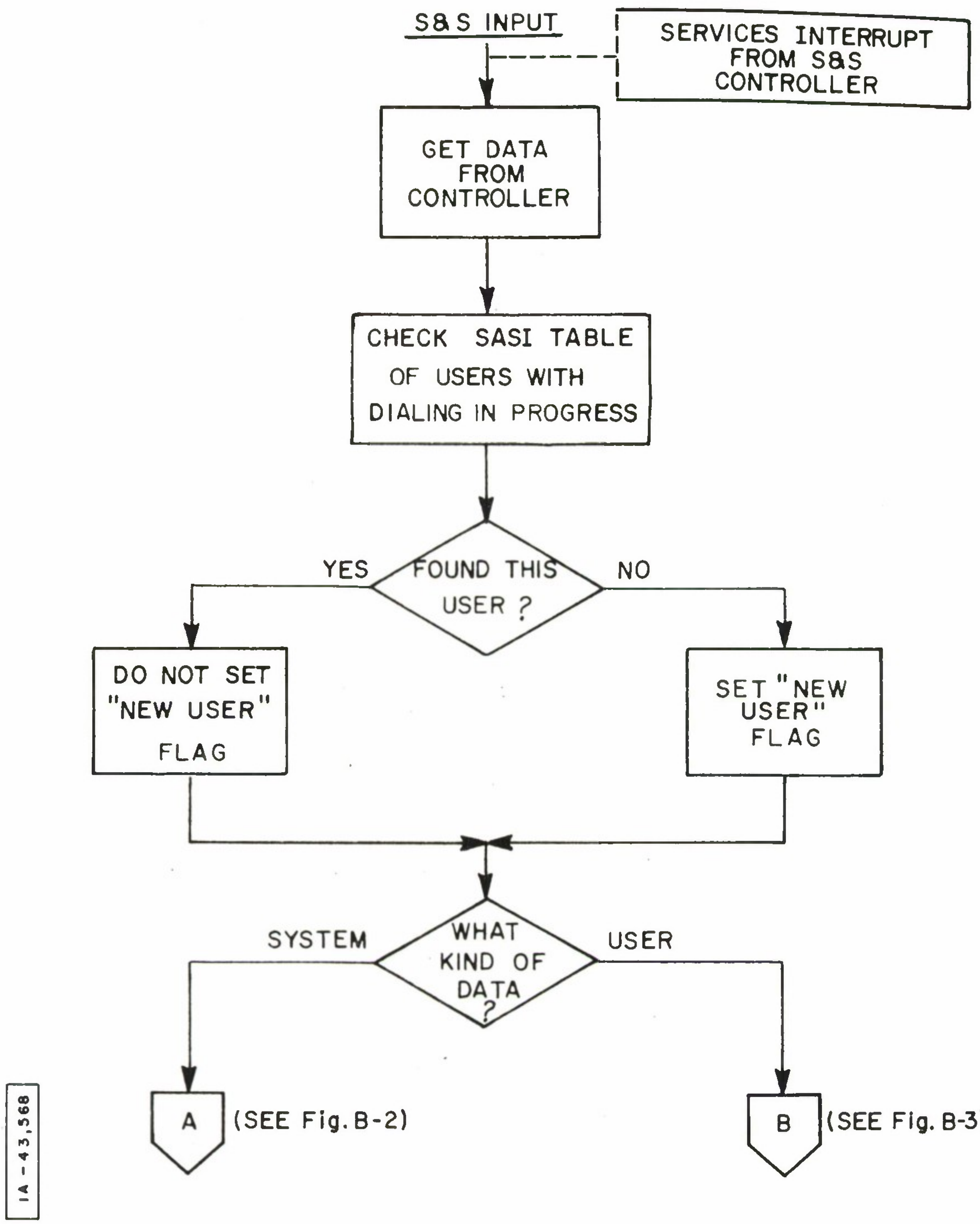


Figure B-1 SIGNALING AND SUPERVISION INPUT(SASI)

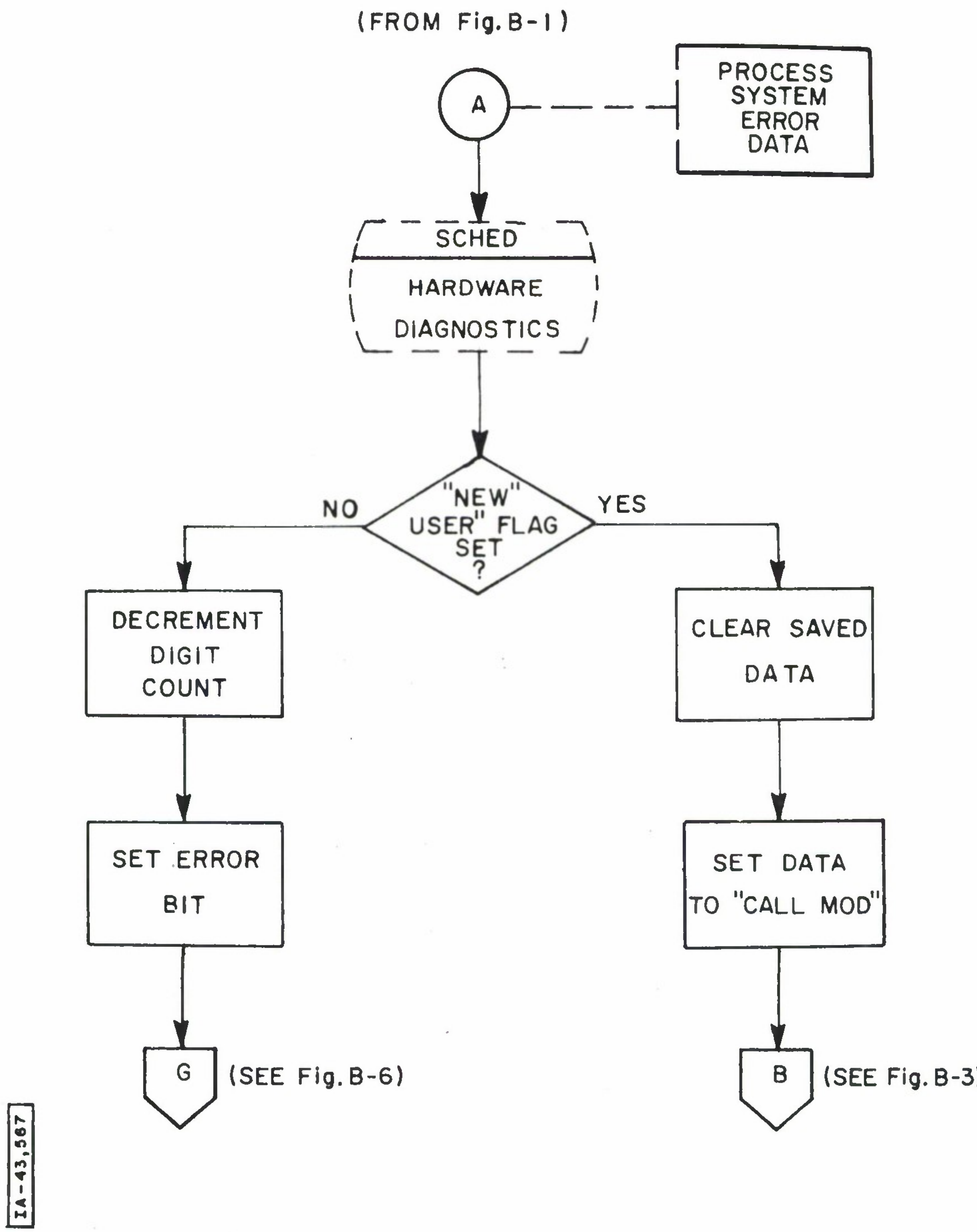


Figure B-2 SIGNALING AND SUPERVISION INPUT (CONT.)

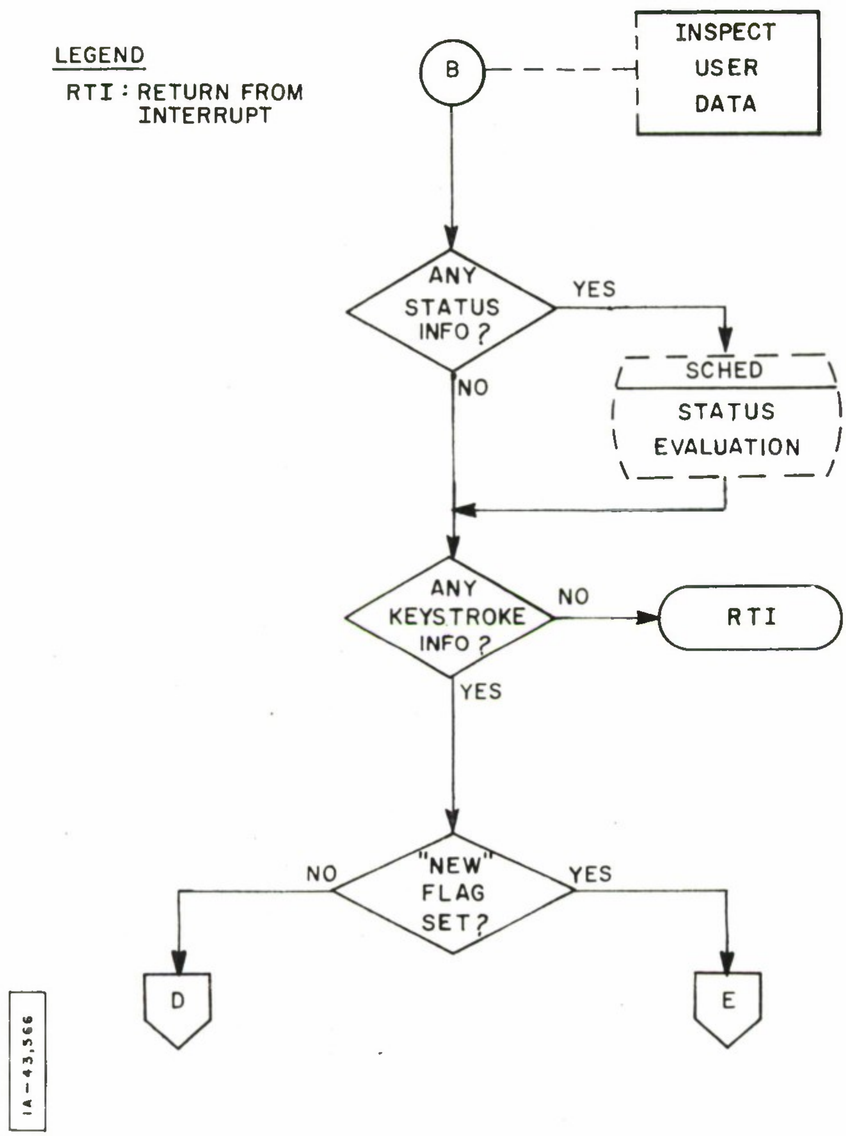


Figure B-3 SIGNALING AND SUPERVISION INPUT (CONT.)

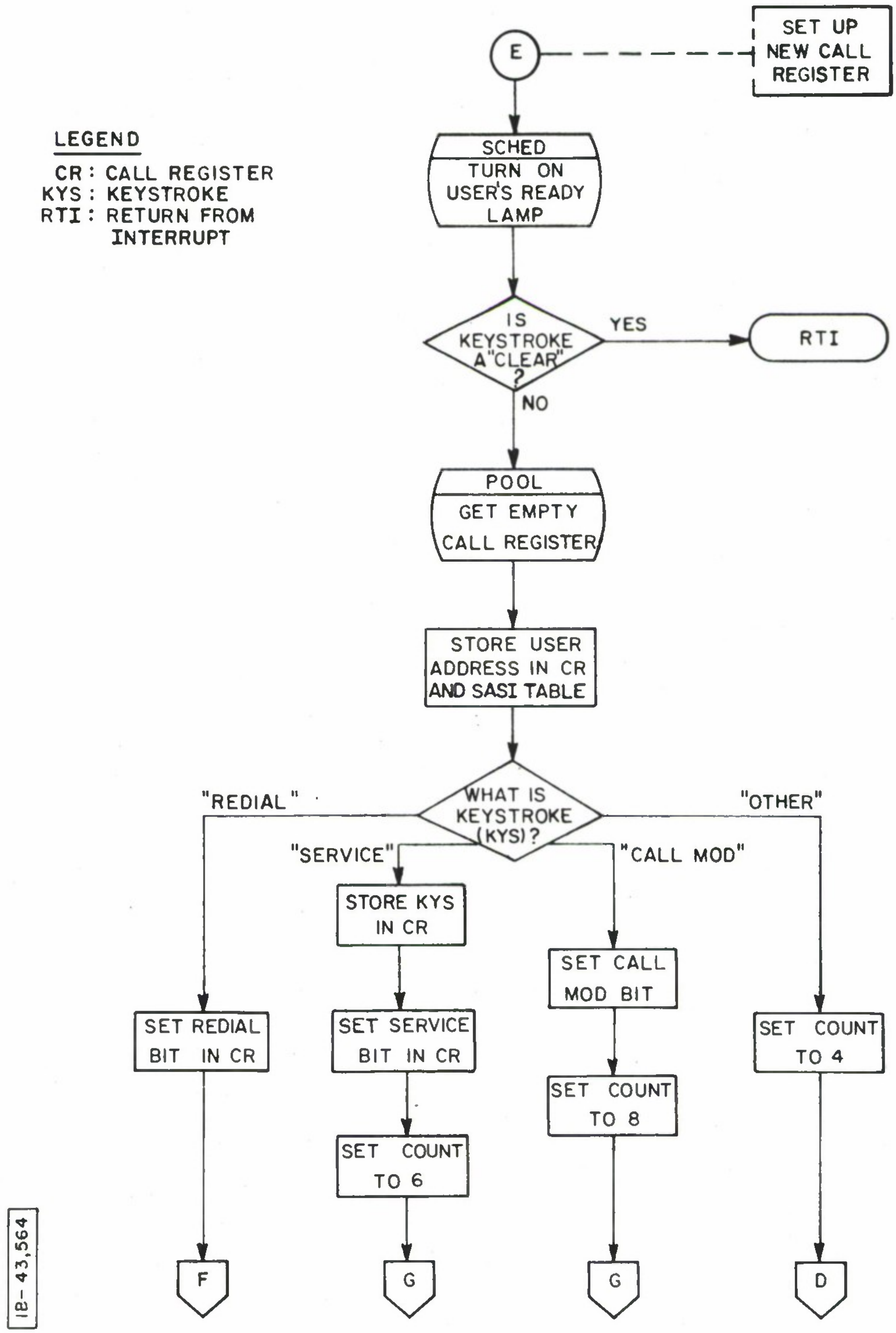


Figure B-4 SIGNALING AND SUPERVISION INPUT (CONT.)

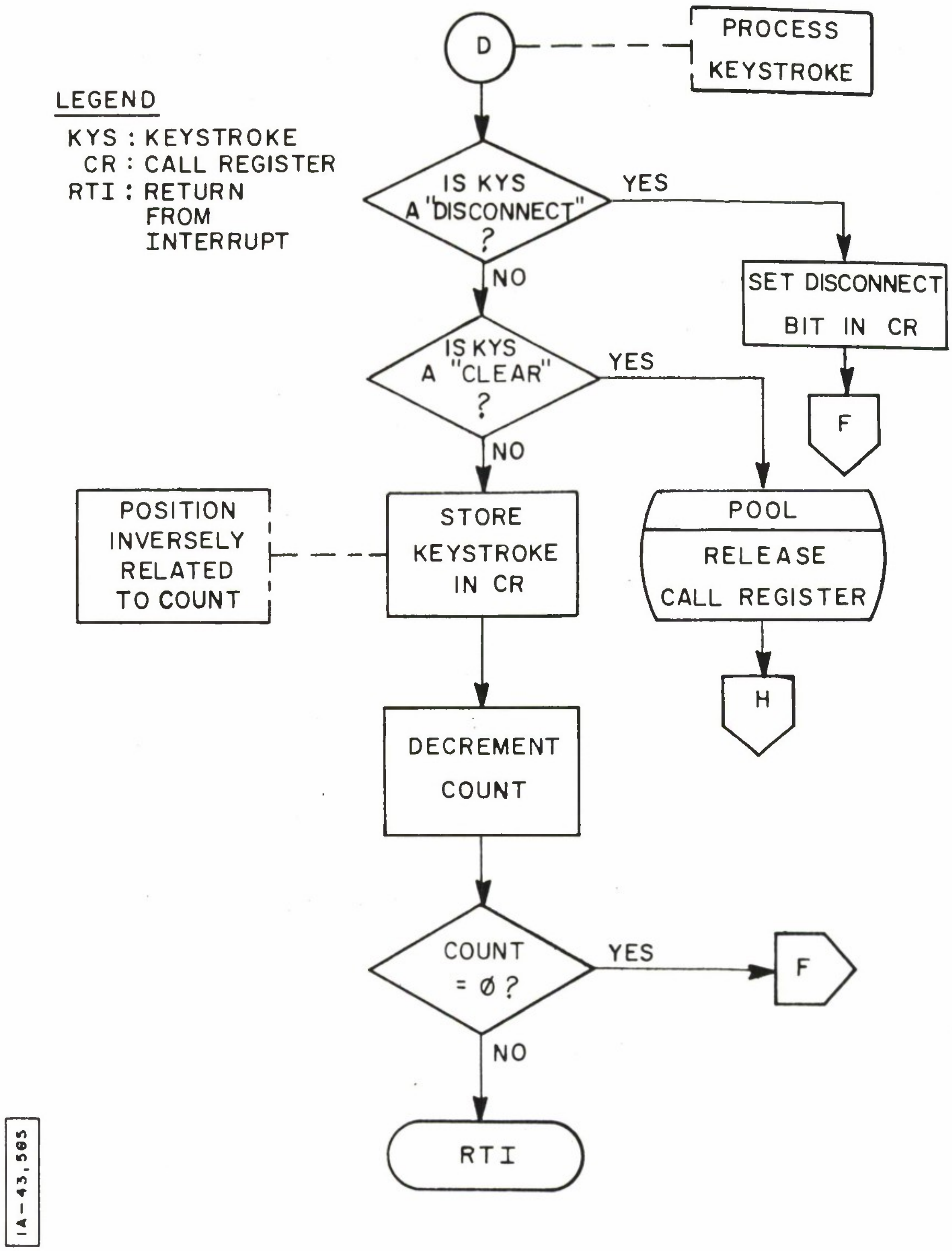
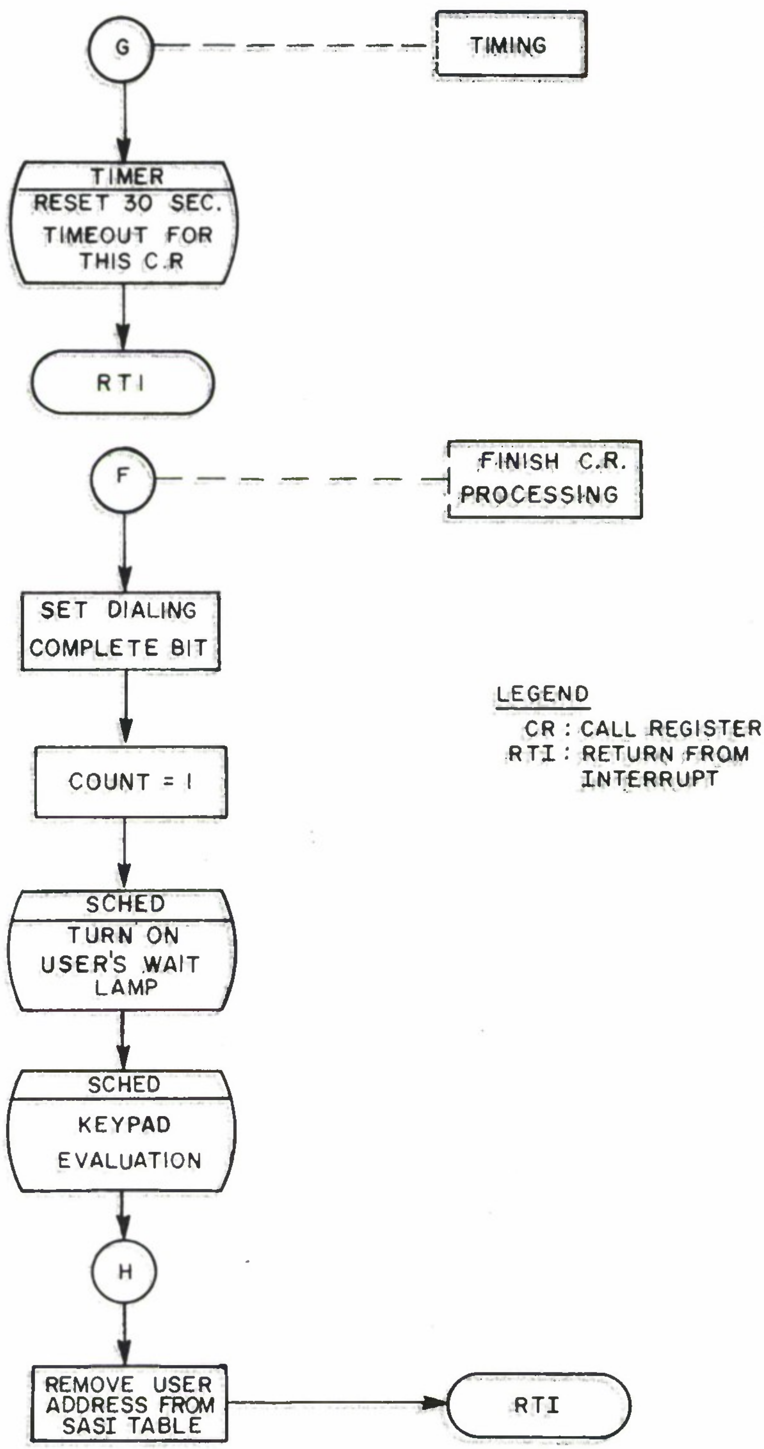


Figure B-5 SIGNALING AND SUPERVISION INPUT (CONT.)



IA-43,572

Figure B-6 SIGNALING AND SUPERVISION INPUT (CONCLUDED)

APPENDIX C

KEYPAD EVALUATION FLOWCHARTS

NOTE:

In accordance with conventional flowcharting techniques, special connector symbols are used to represent an entry from, or an exit to, another part of the program flowchart.

The symbol is used to indicate an exit from a page. The letter within the symbol indicates the entry point that will be found on another page.

The symbol ○ is used to indicate an exit to another part of the program on the same page. It is also used in the following flowcharts to indicate the entry point from another page.

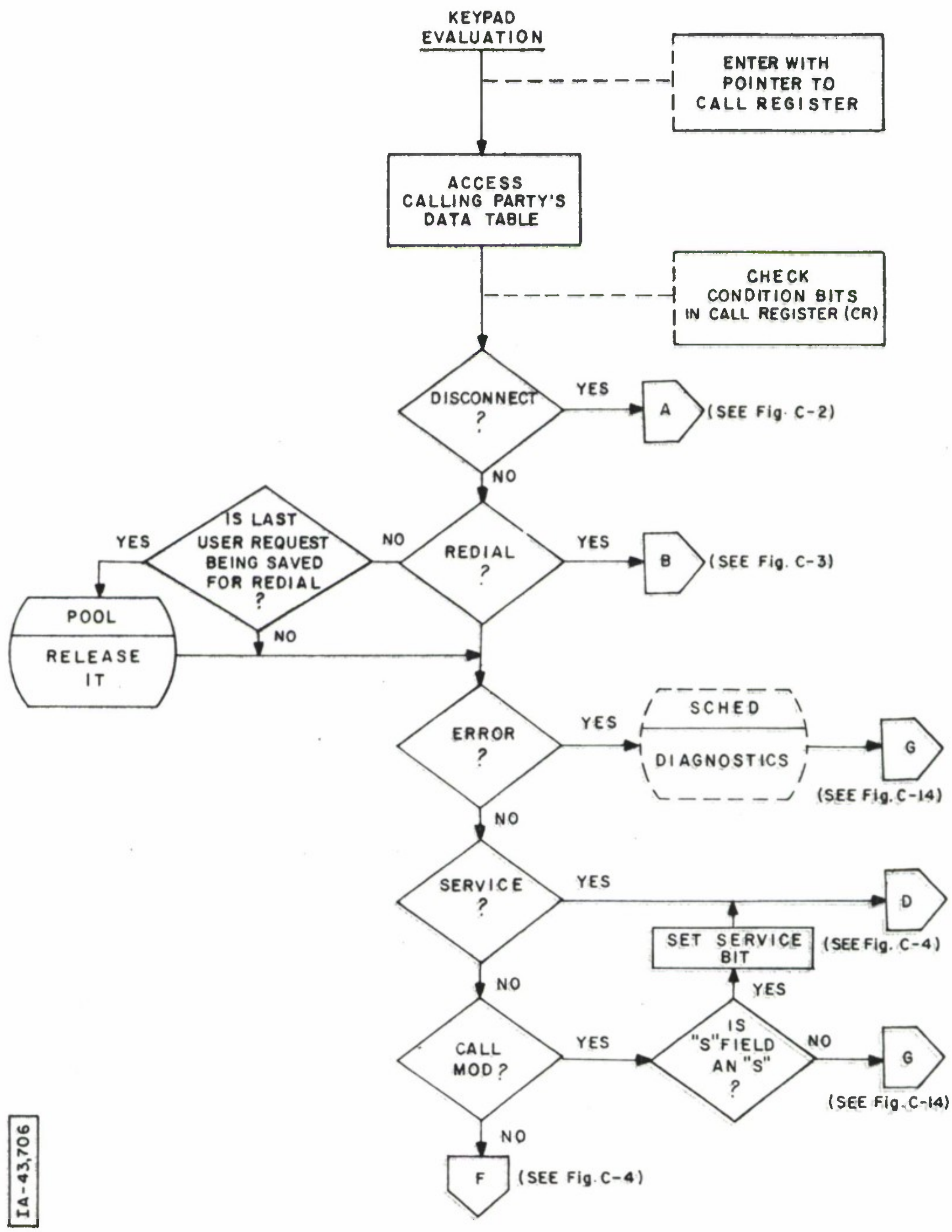


Figure C-1 KEYPAD EVALUATION

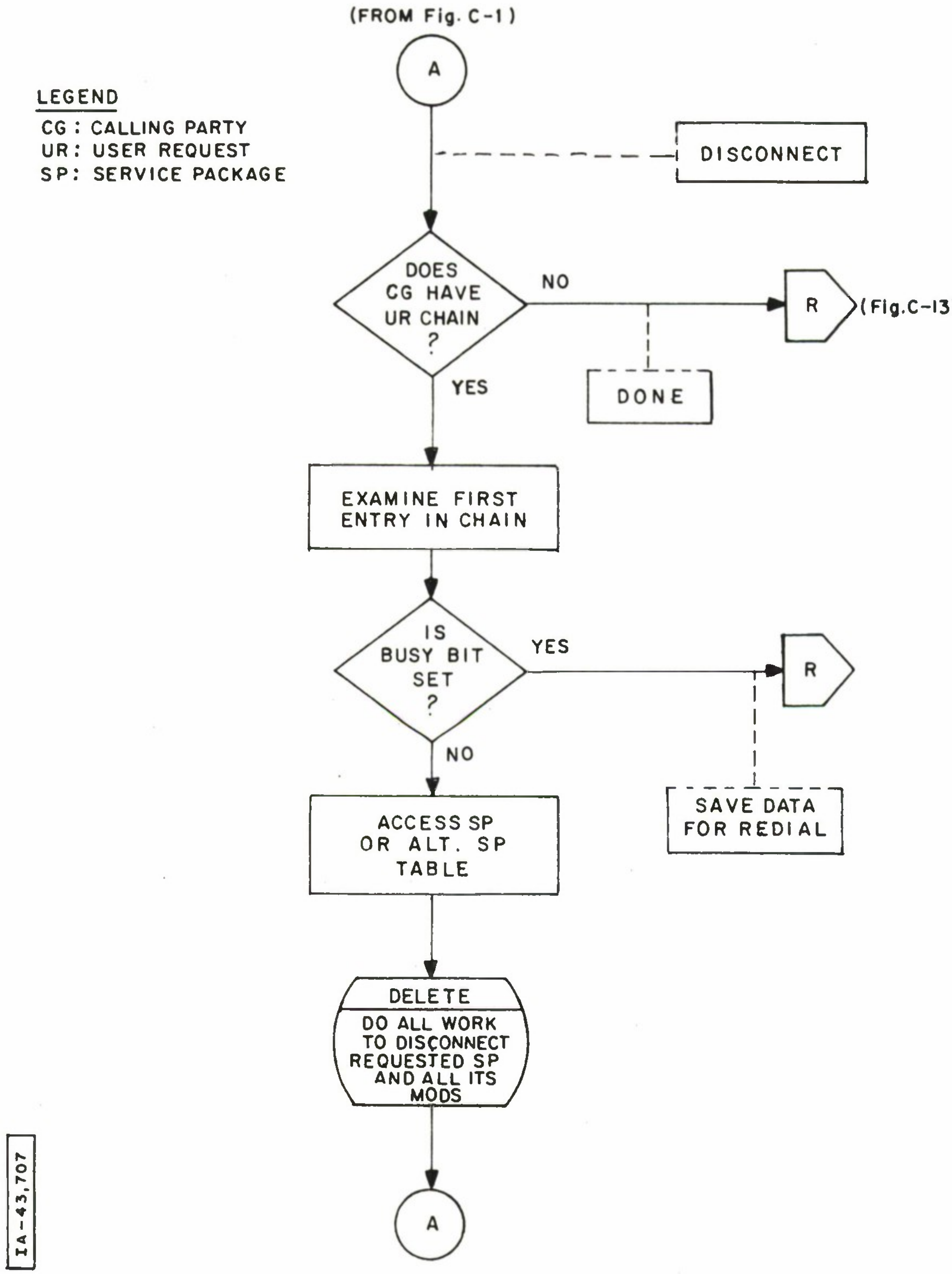
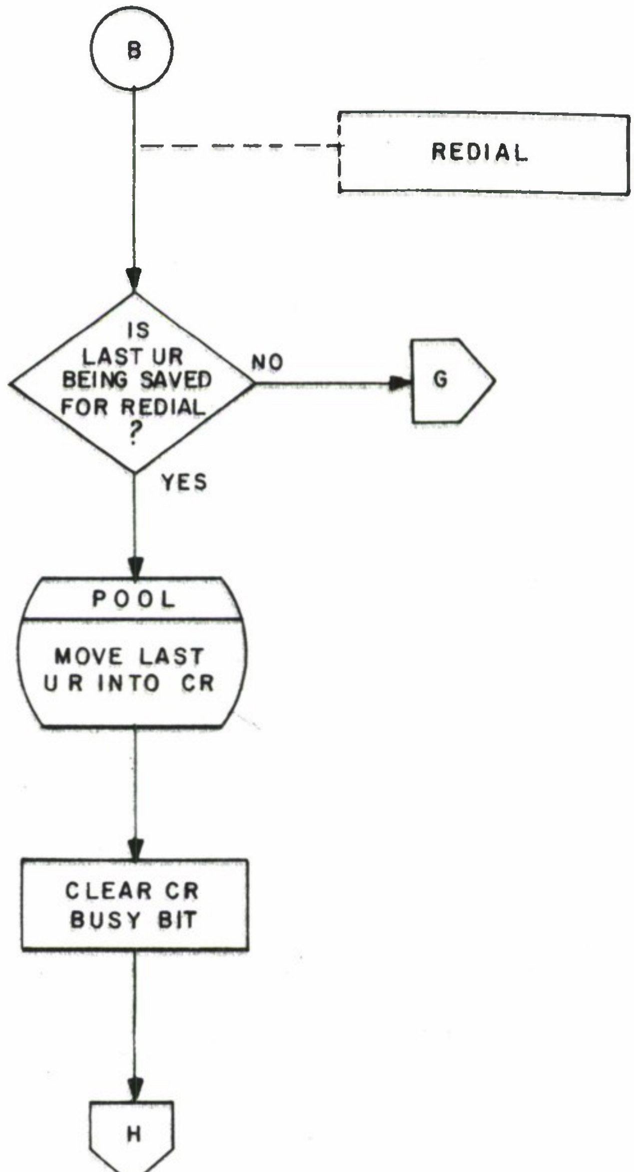


Figure C-2 KEYPAD EVALUATION (CONTINUED)

LEGEND

UR: USER REQUEST
CR: CALL REGISTER



IA-43,708

Figure C-3 KEYPAD EVALUATION (CONTINUED)

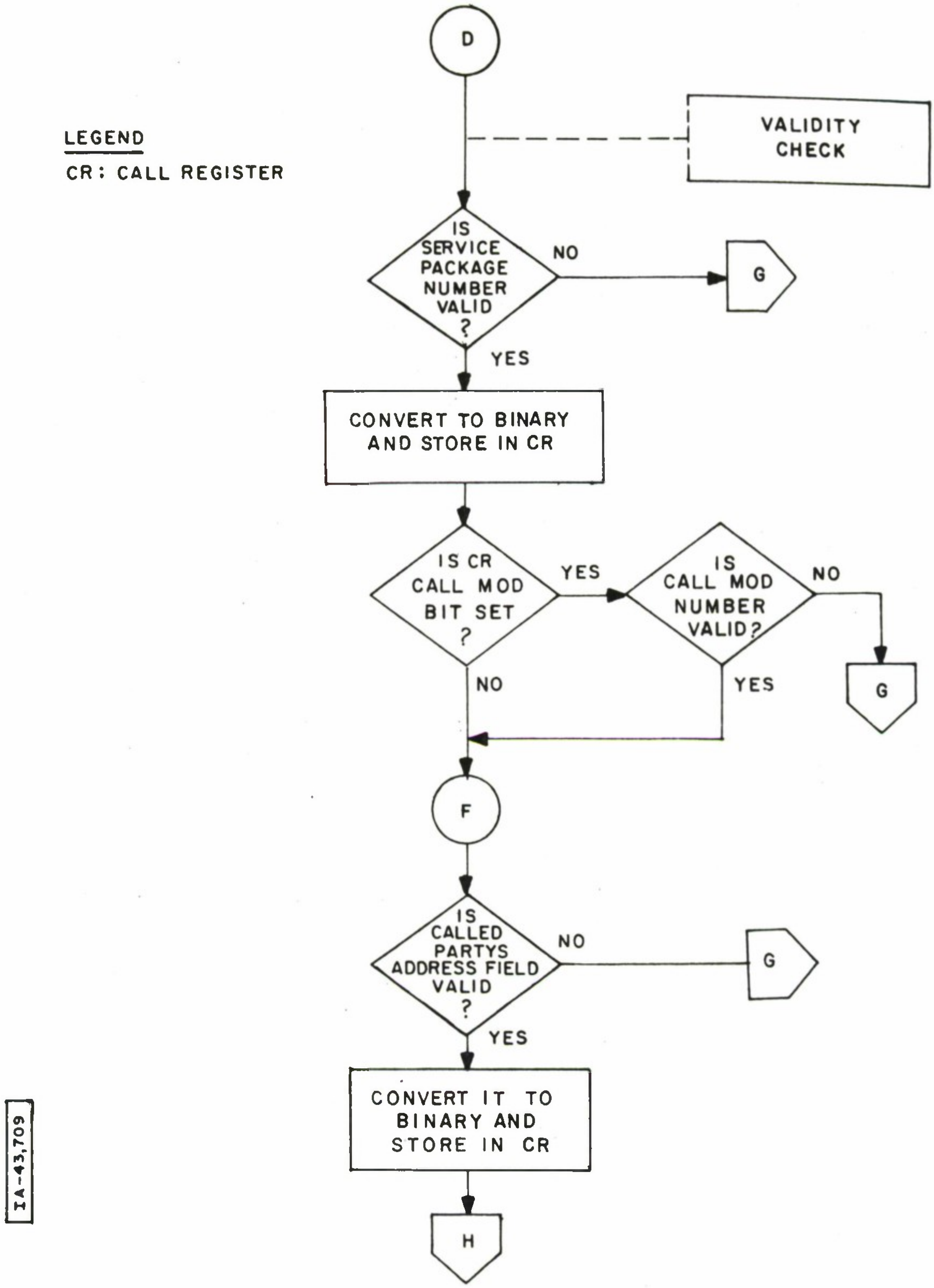


Figure C-4 KEYPAD EVALUATION (CONTINUED)

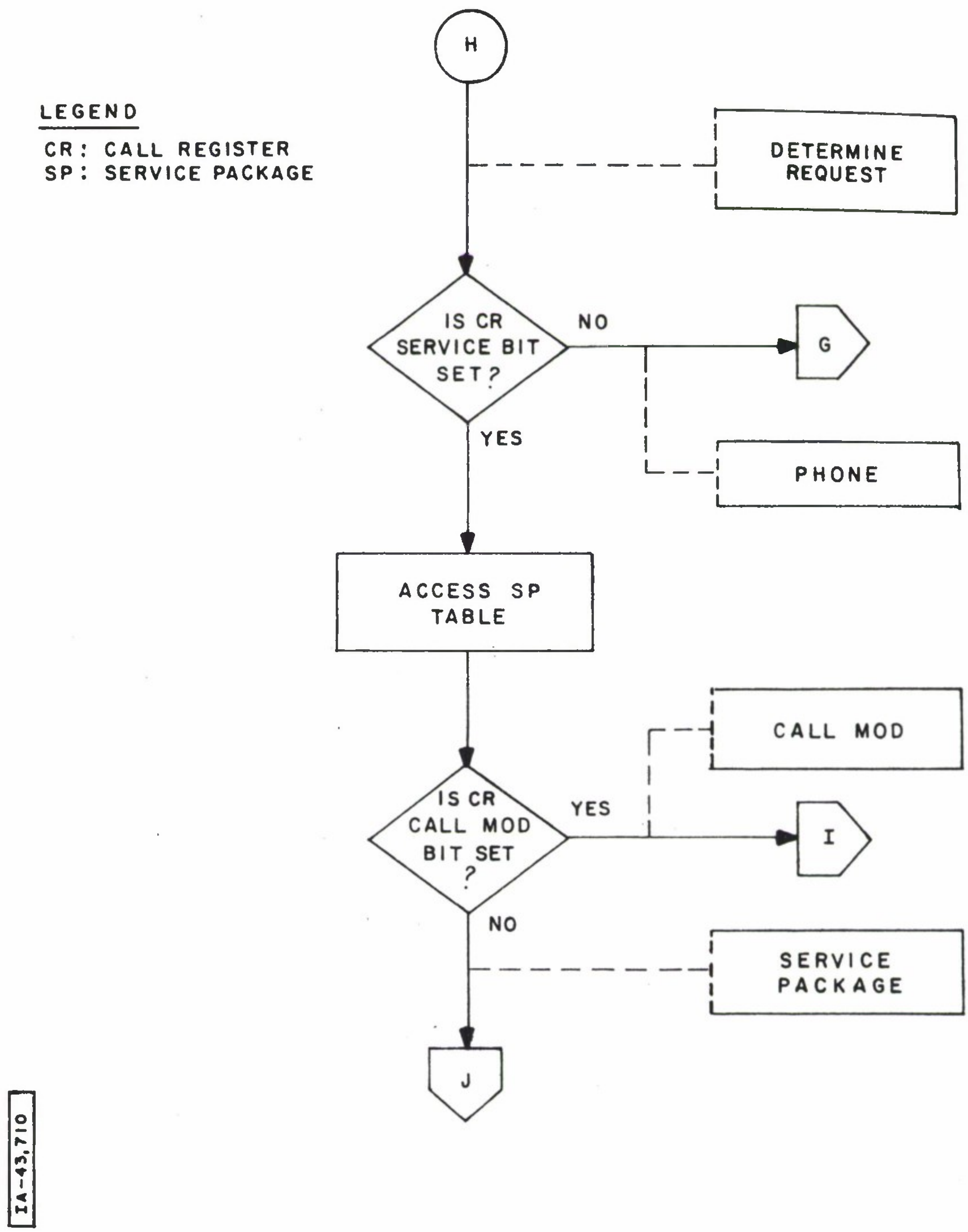


Figure C-5 KEYPAD EVALUATION (CONTINUED)

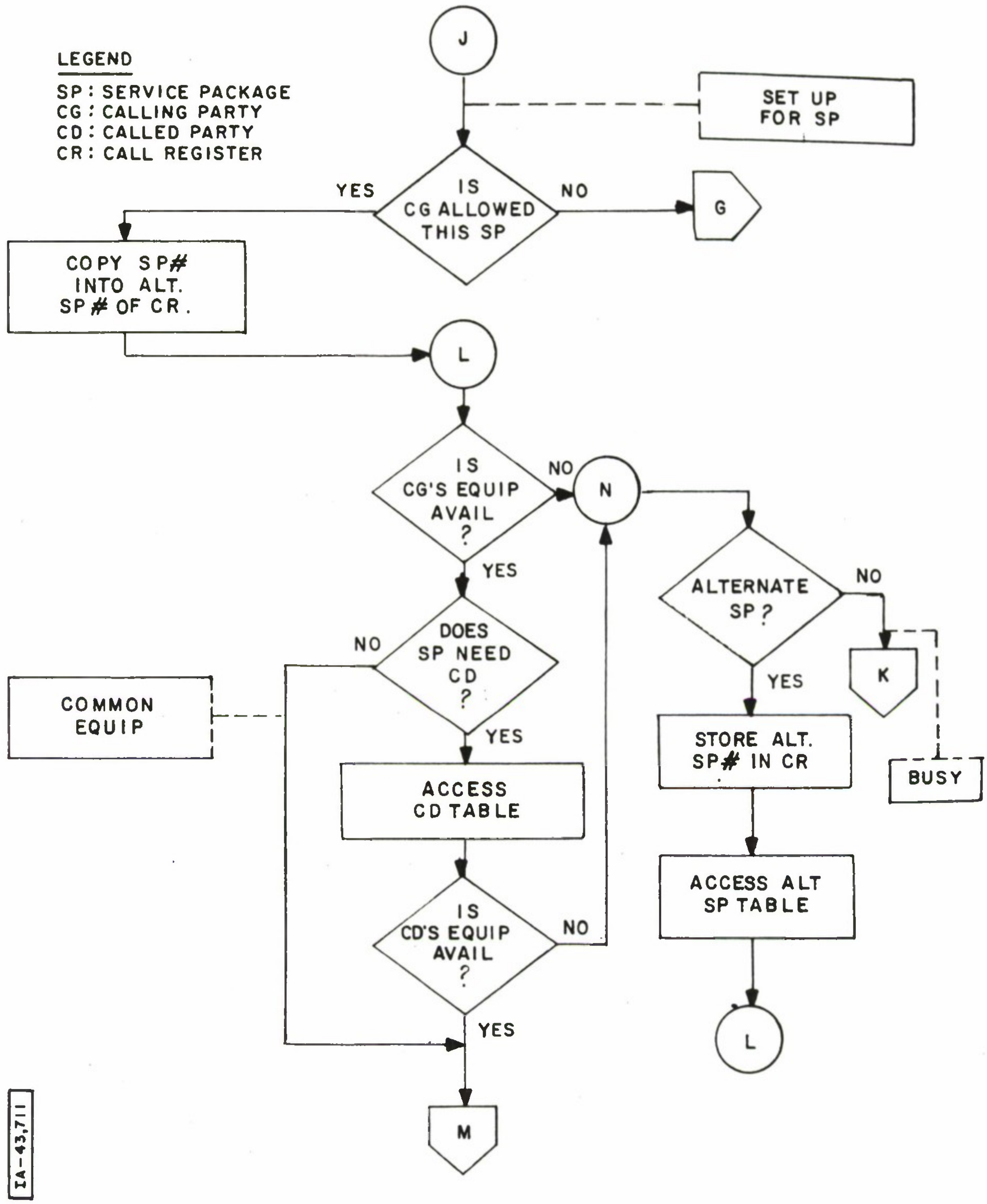


Figure C-6 KEYPAD EVALUATION (CONTINUED)

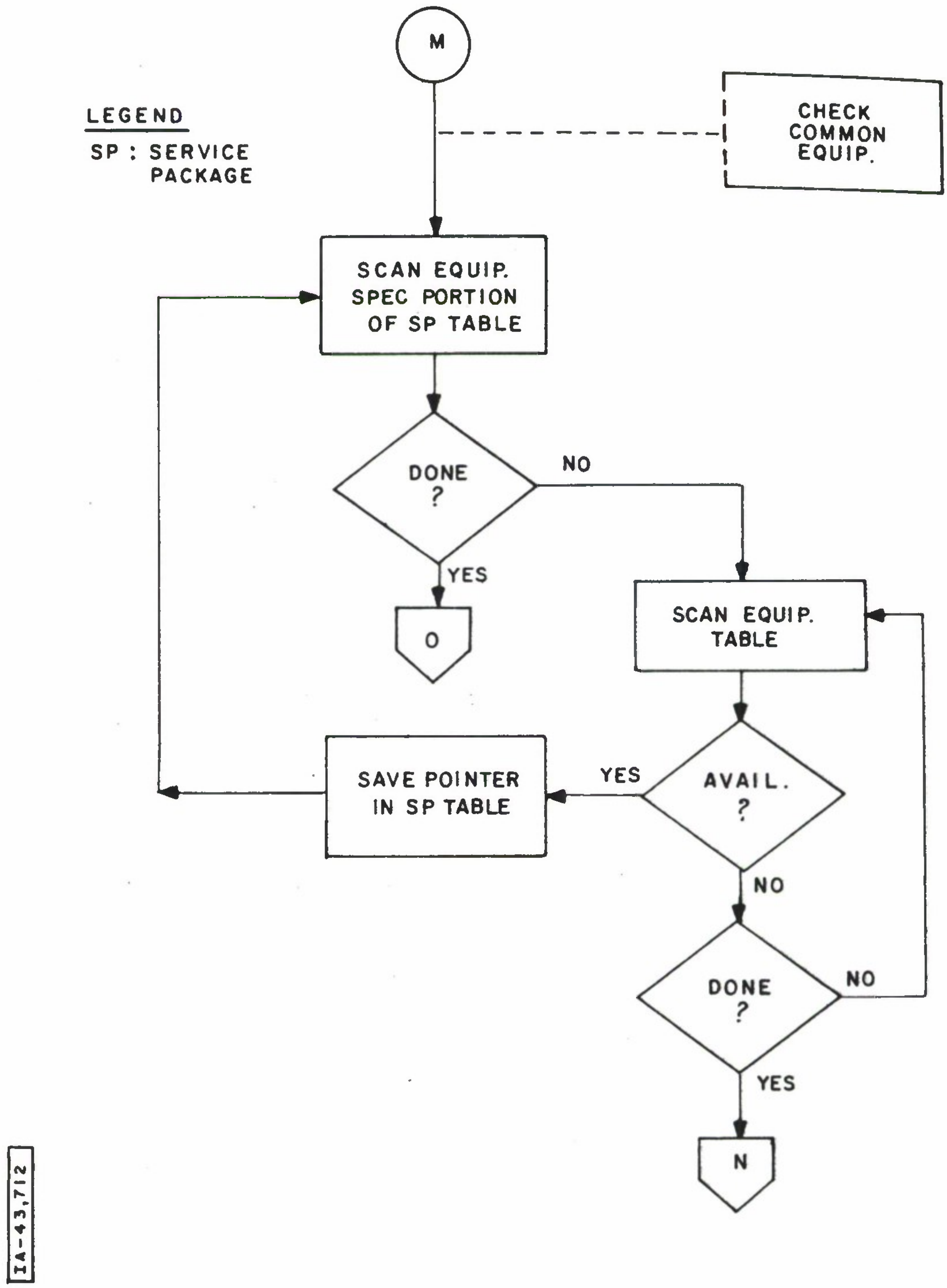
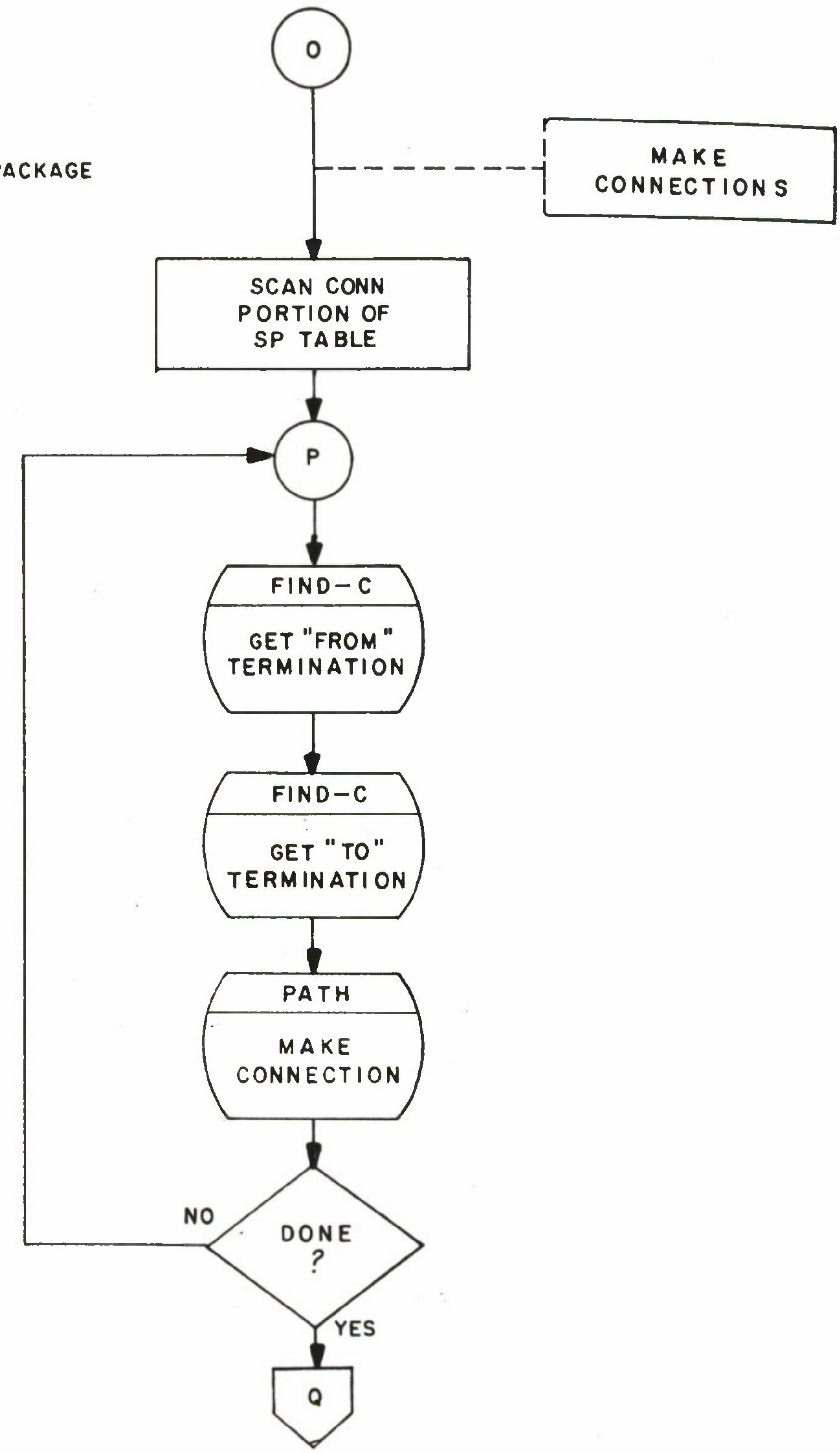


Figure C-7 KEYPAD EVALUATION (CONTINUED)

LEGEND

SP : SERVICE PACKAGE



IA-43,713

Figure C-8 KEYPAD EVALUATION (CONTINUED)

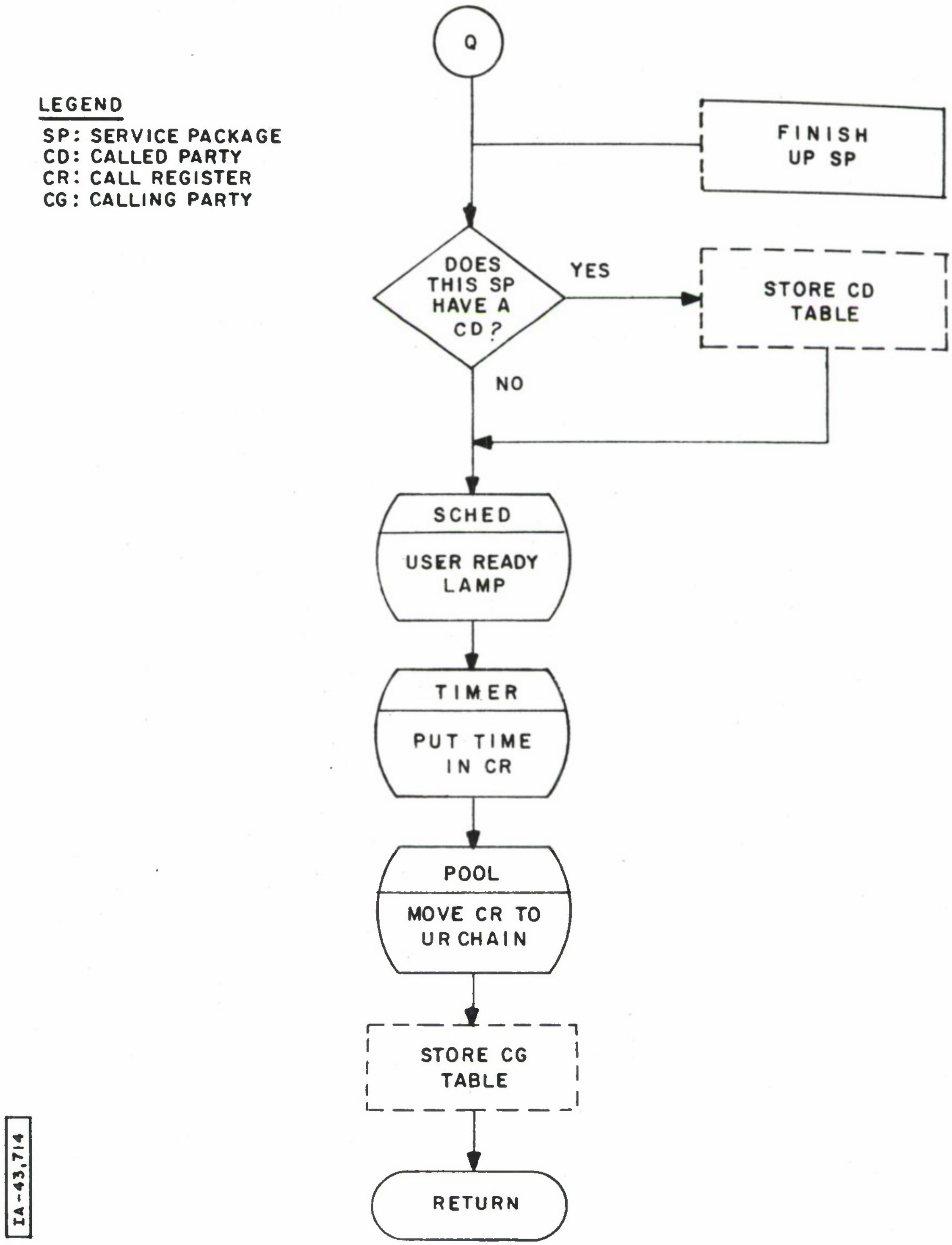


Figure C-9 KEYPAD EVALUATION (CONTINUED)

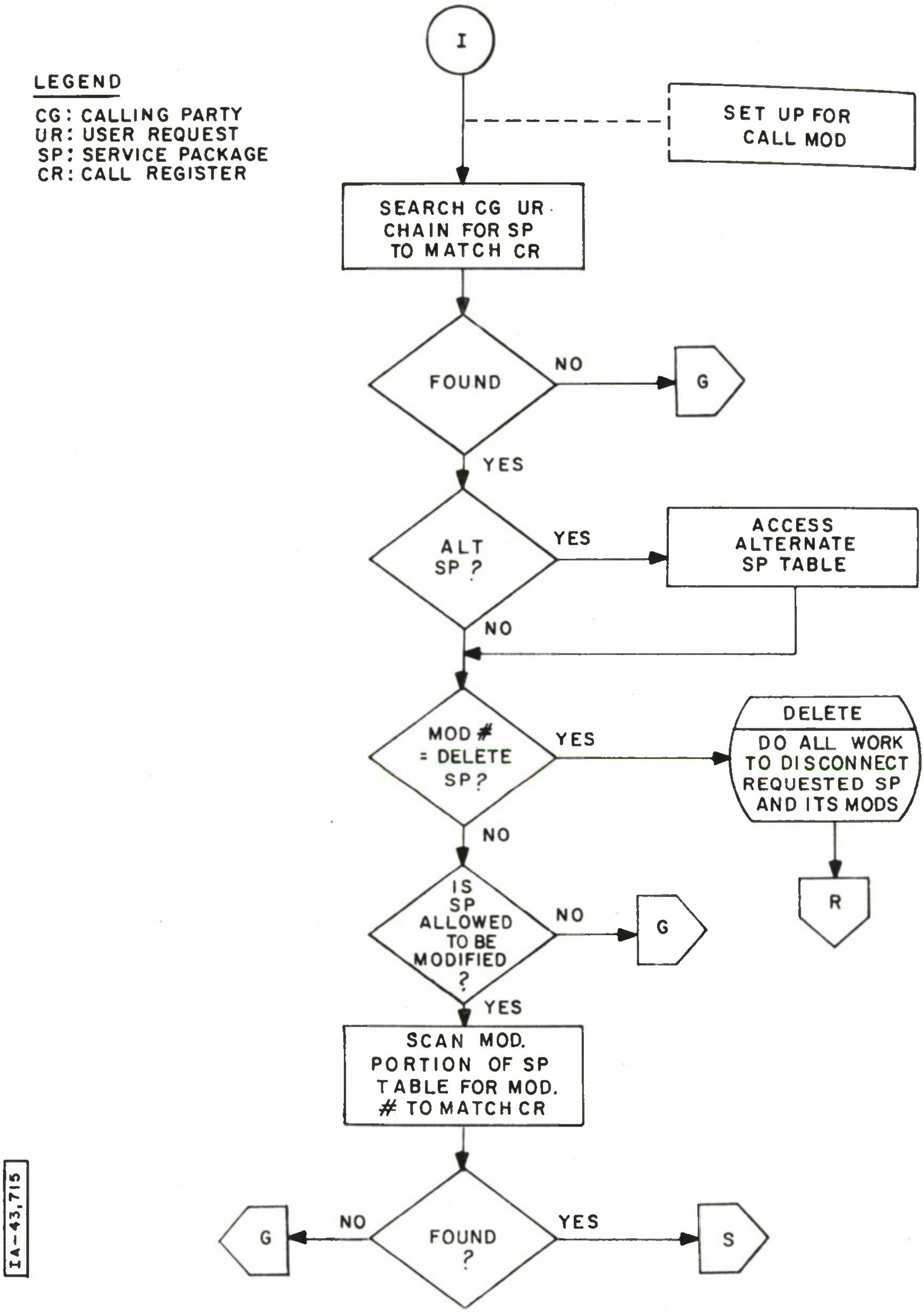


Figure C-10 KEYPAD EVALUATION (CONTINUED)

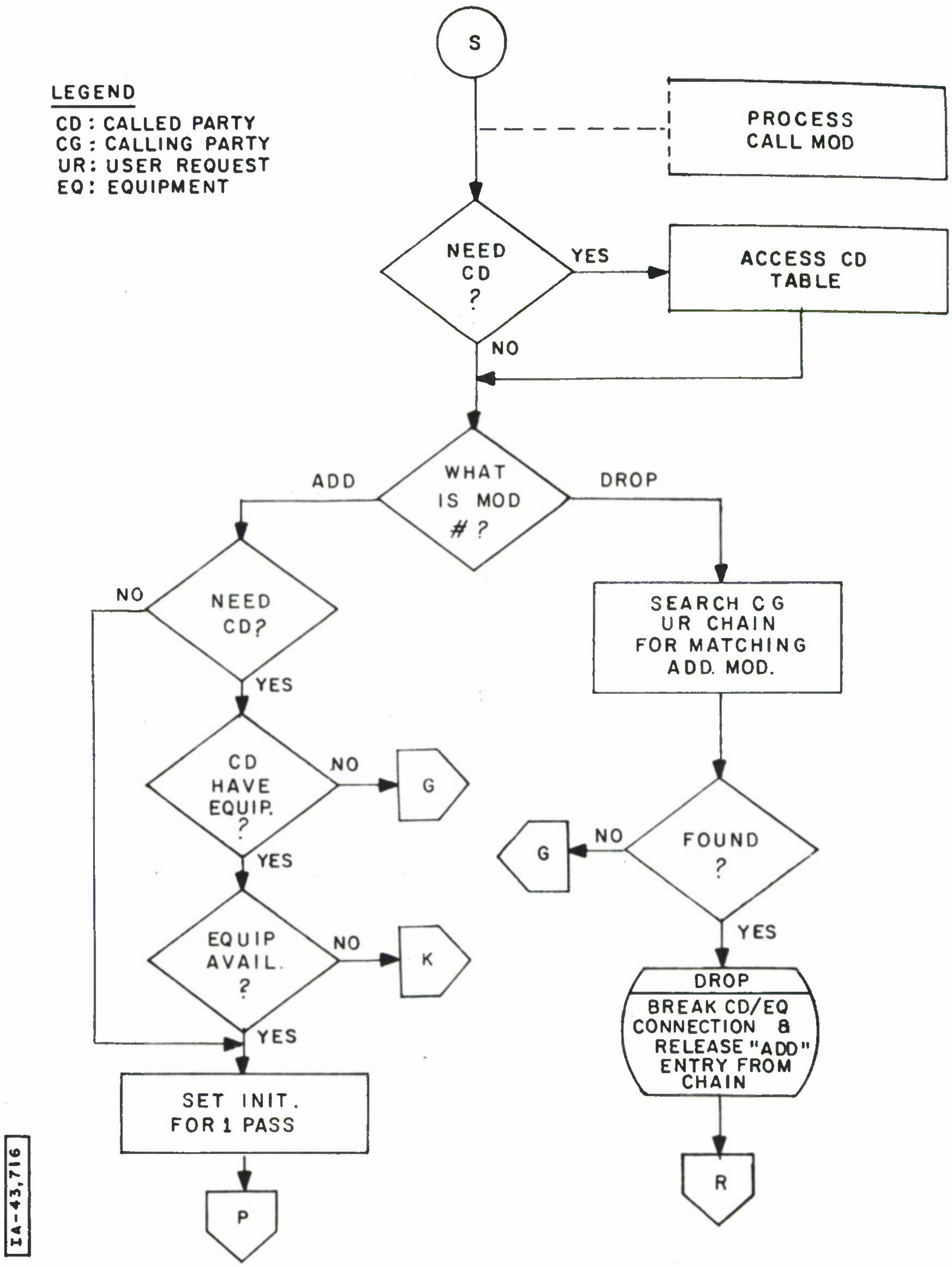
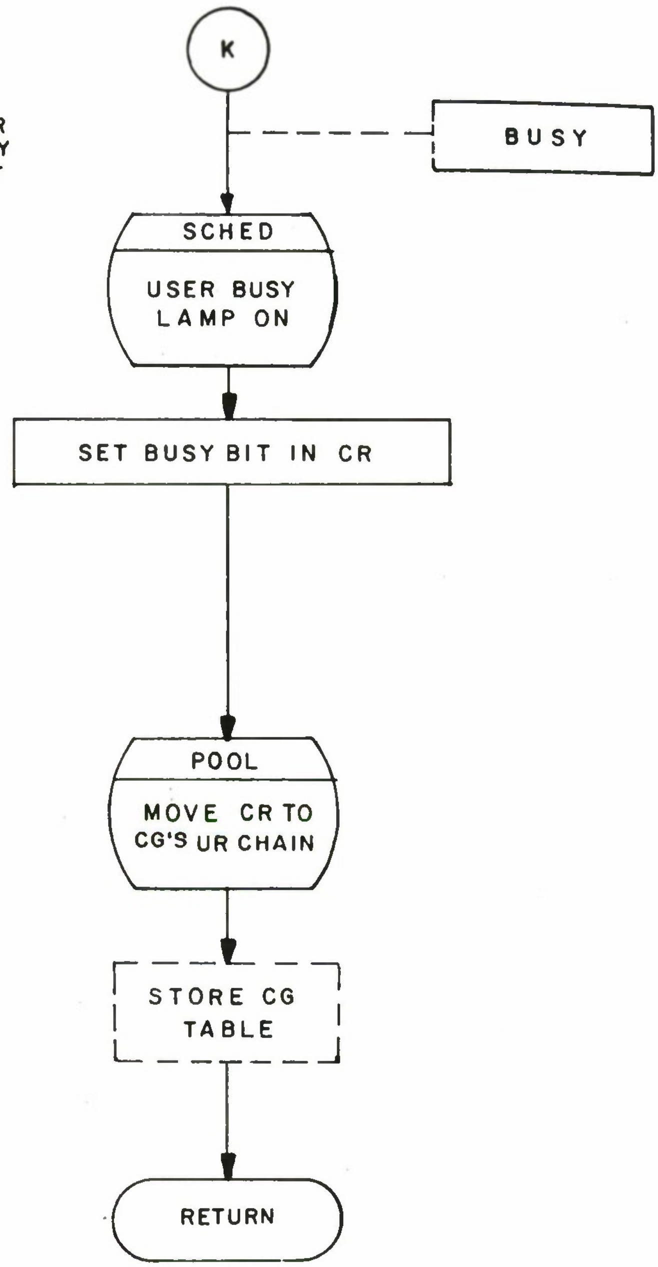


Figure C-11 KEYPAD EVALUATION (CONTINUED)

LEGEND

CR : CALL REGISTER
CG : CALLING PARTY
UR : USER REQUEST



IA-43,717

Figure C-12 KEYPAD EVALUATION (CONTINUED)

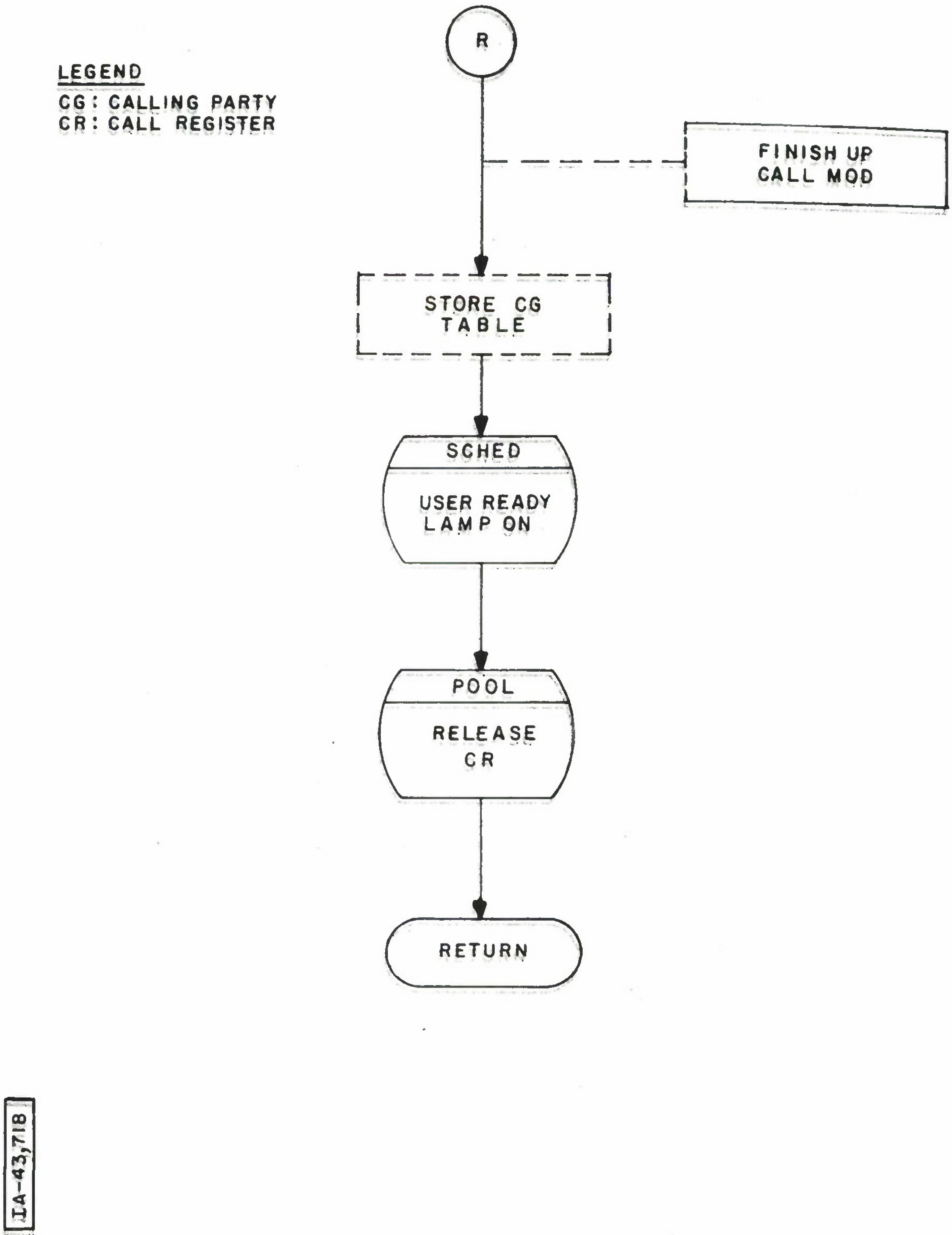


Figure C-13 KEYPAD EVALUATION (CONTINUED)

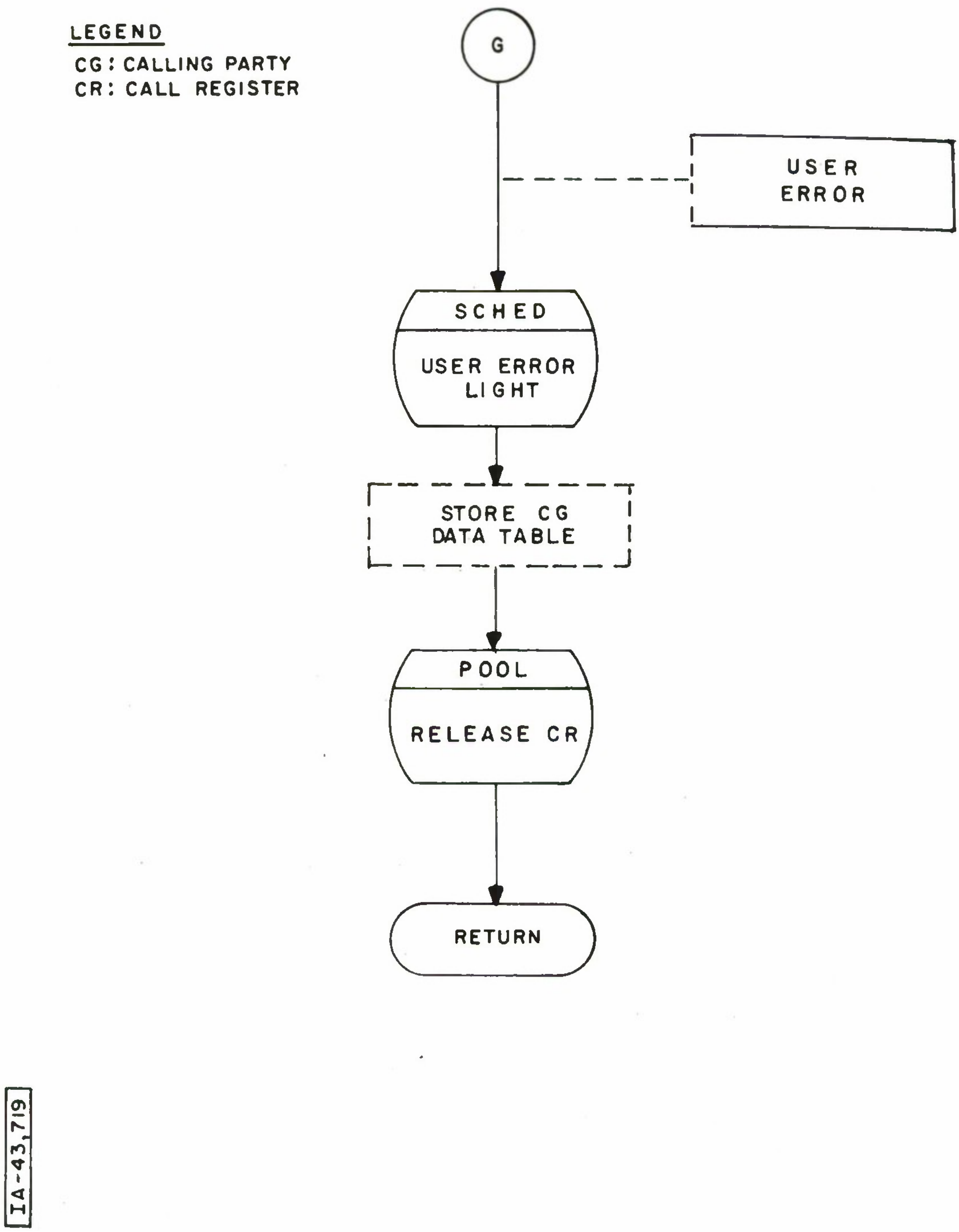


Figure C-14 KEYPAD EVALUATION (CONCLUDED)

APPENDIX D

PATH SELECTION FLOWCHARTS

NOTE:

In accordance with conventional flowcharting techniques, special connector symbols are used to represent an entry from, or an exit to, another part of the program flowchart.

The symbol is used to indicate an exit from a page. The letter within the symbol indicates the entry point that will be found on another page.

The symbol is used to indicate an exit to another part of the program on the same page. It is also used in the following flowcharts to indicate the entry point from another page.

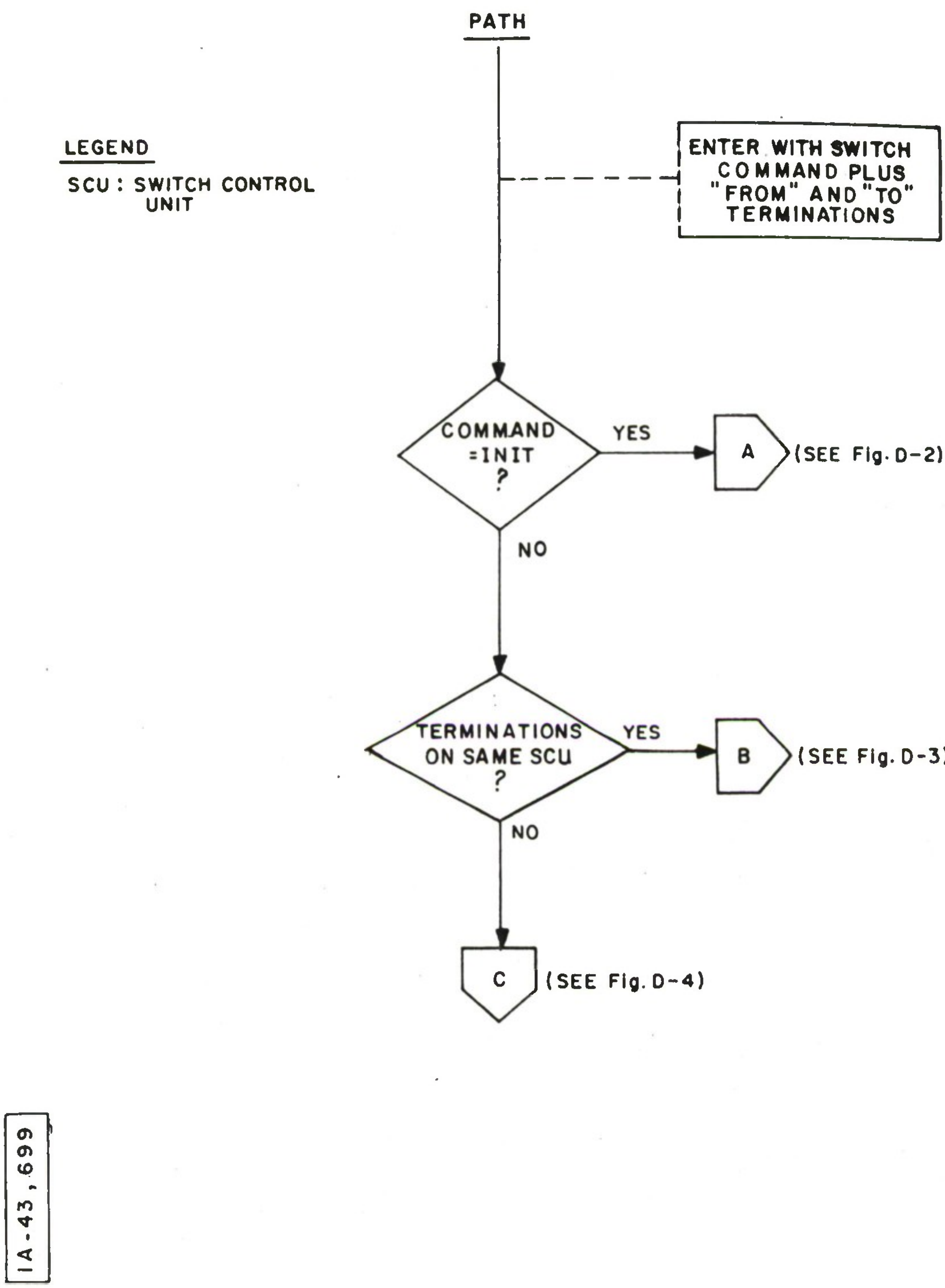


Figure D-1 PATH SELECTION

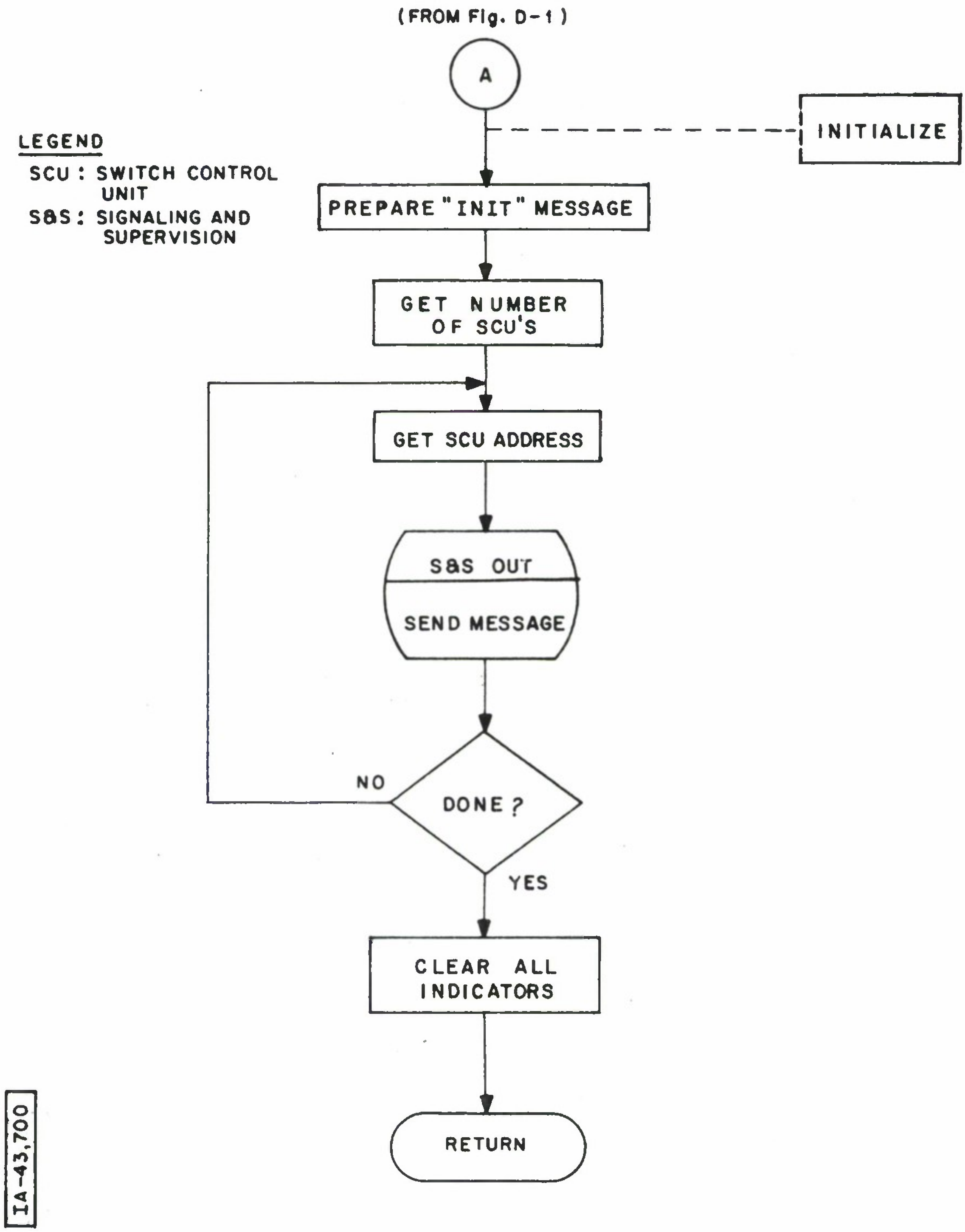


Figure D-2 PATH SELECTION (CONTINUED)

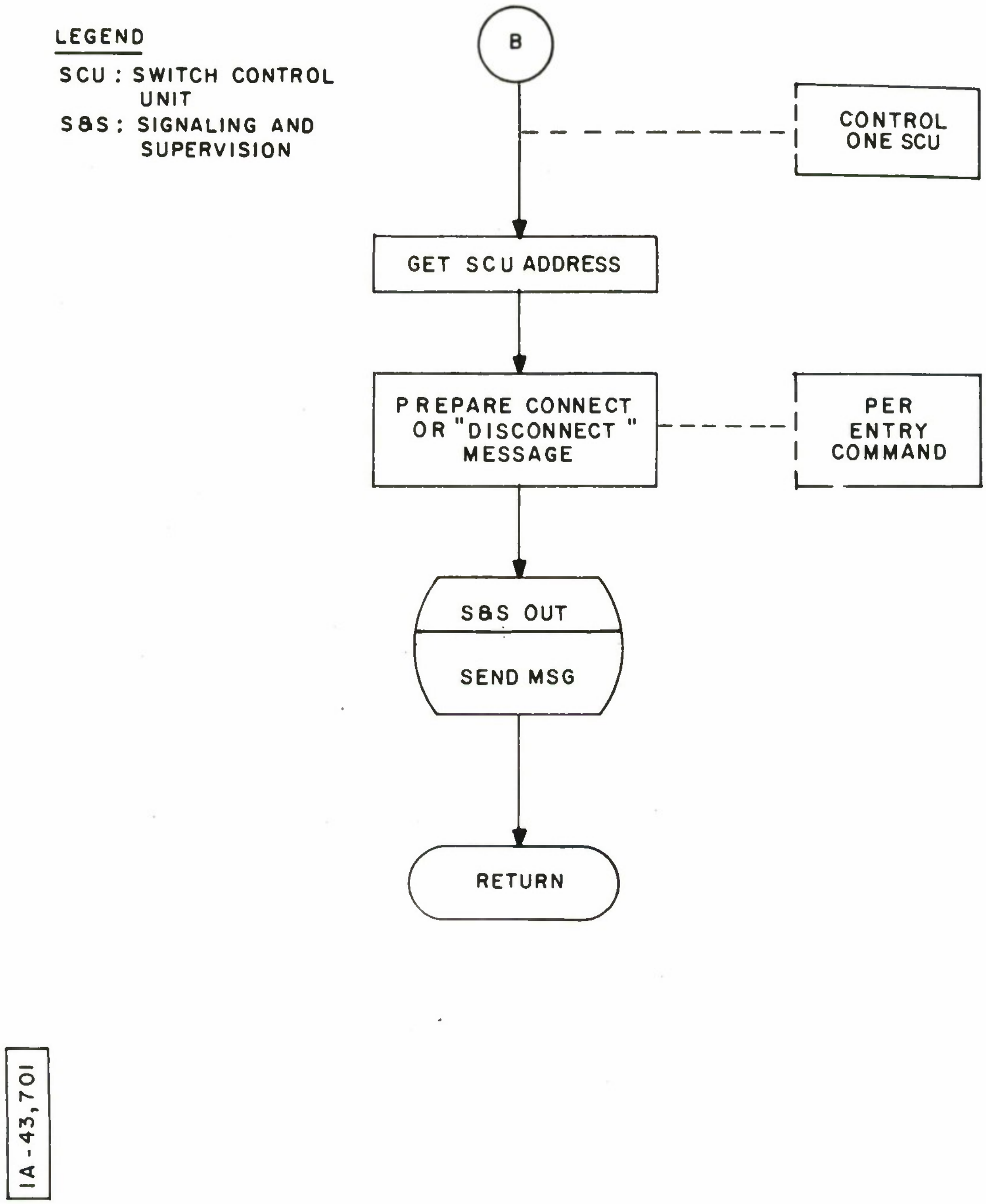


Figure D-3 PATH SELECTION (CONT)

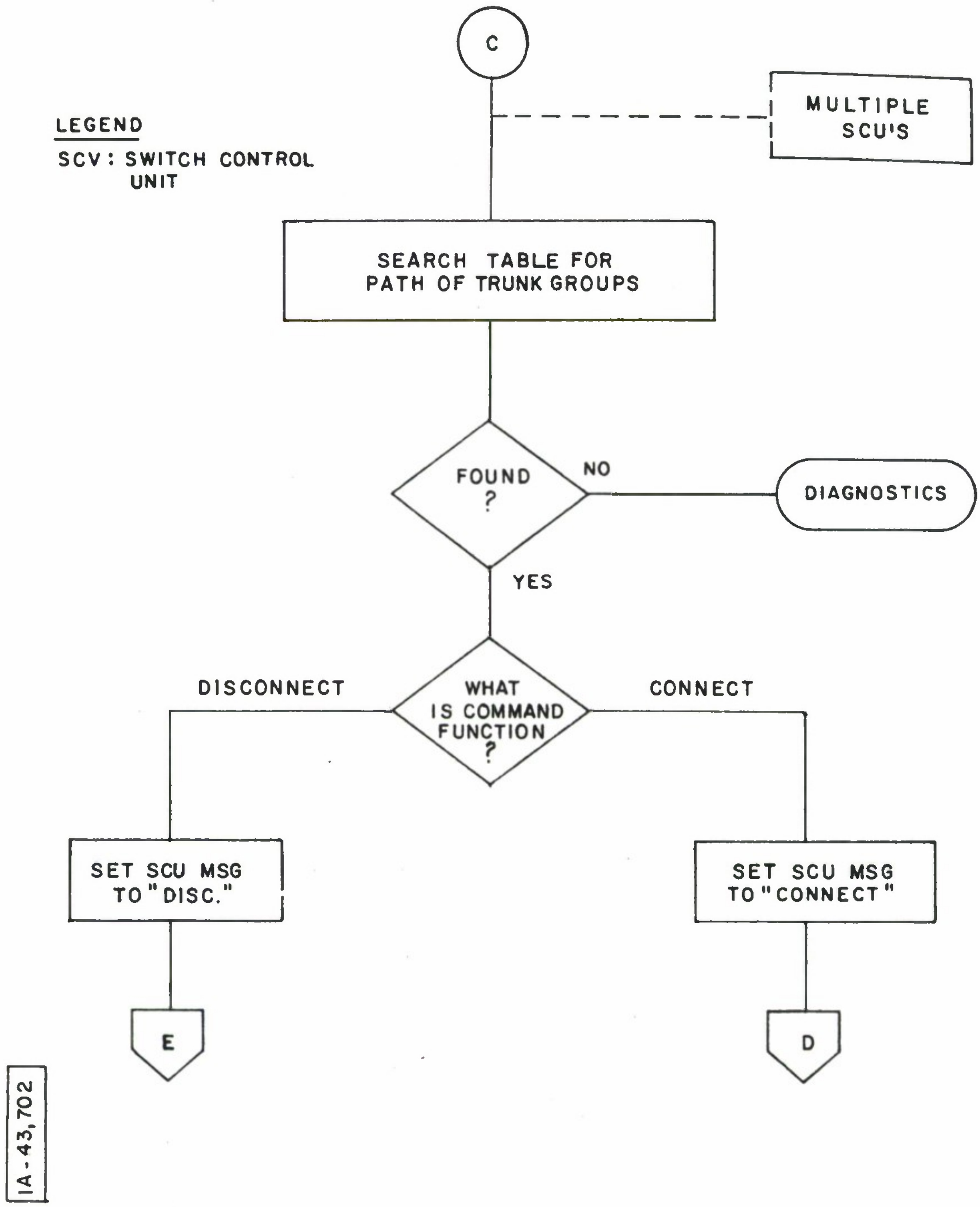
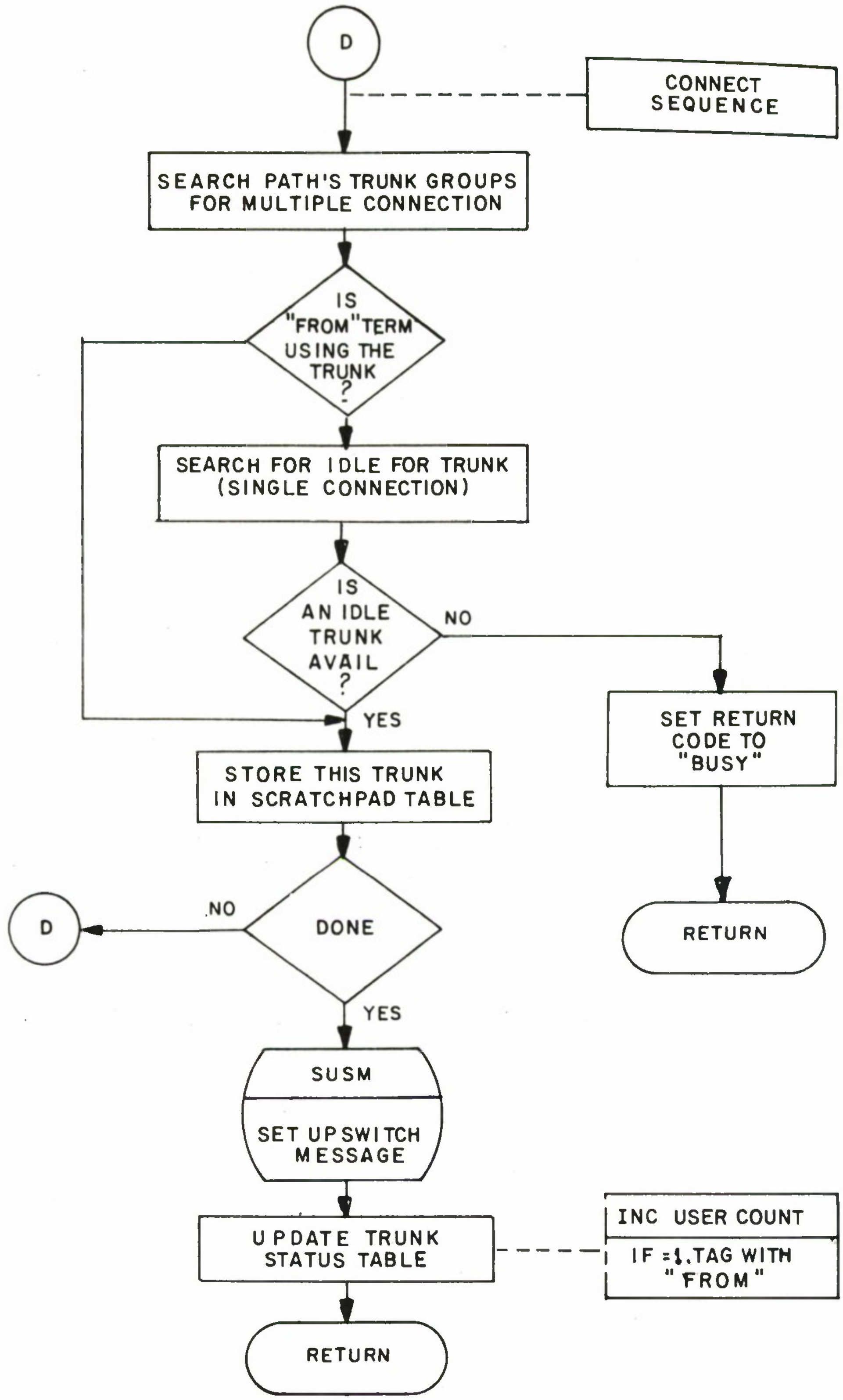
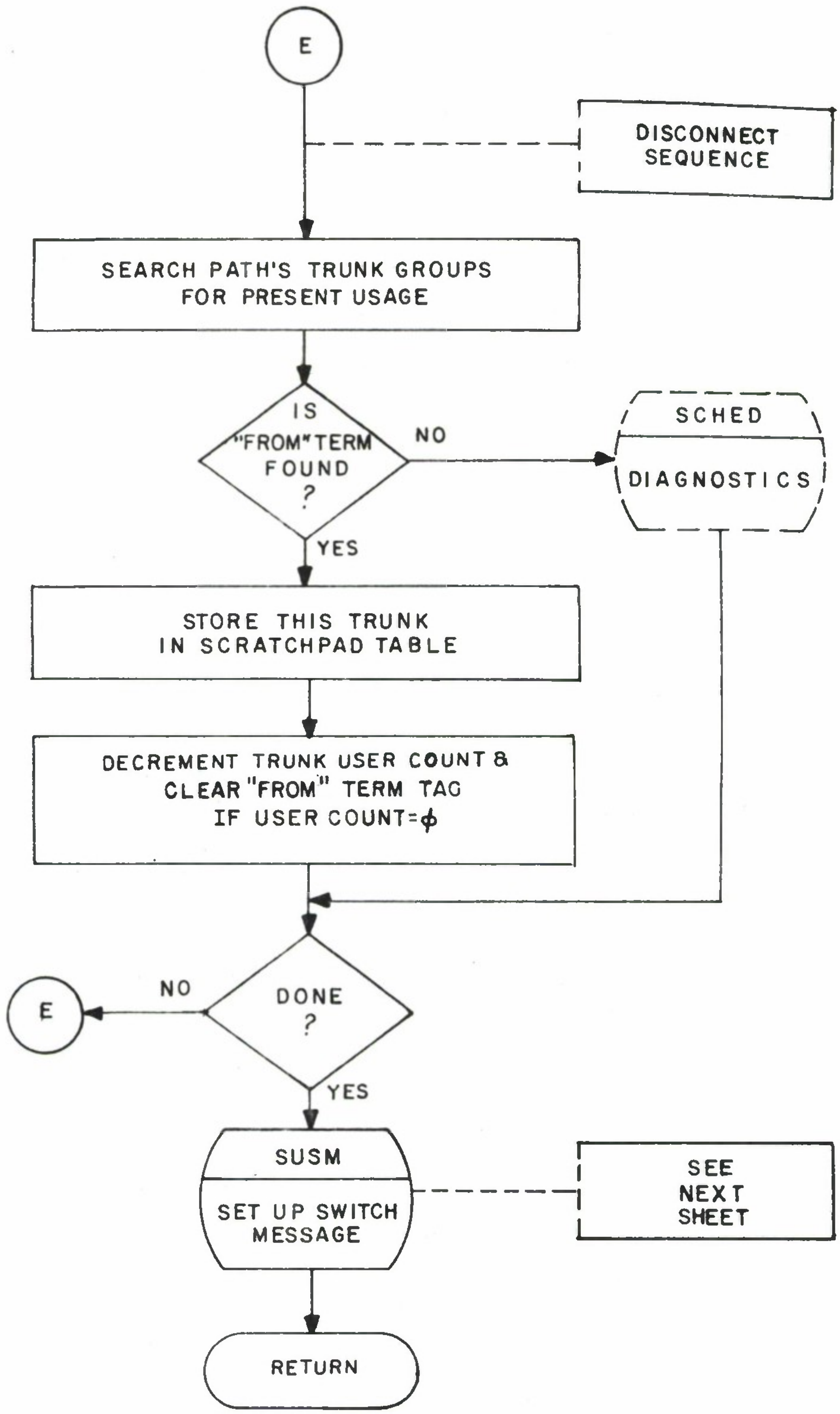


Figure D-4 PATH SELECTION (CONT.)



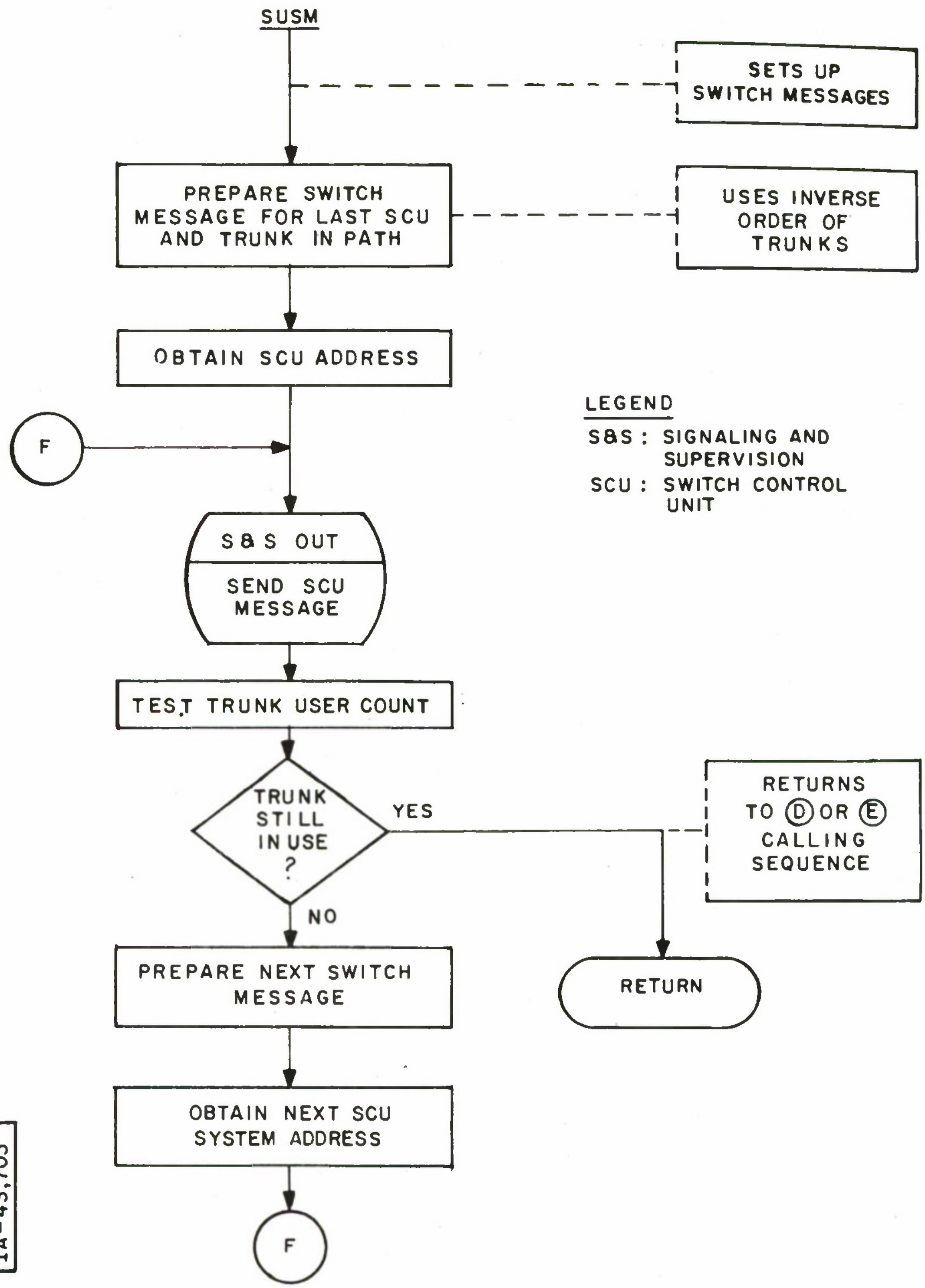
IA-43,703

Figure D-5 PATH SELECTION (CONTINUED)



IA-43,704

Figure D-6 PATH SELECTION (CONT.)



IA-43,705

Figure D-7 PATH SELECTIONS (CONCLUDED)

APPENDIX E

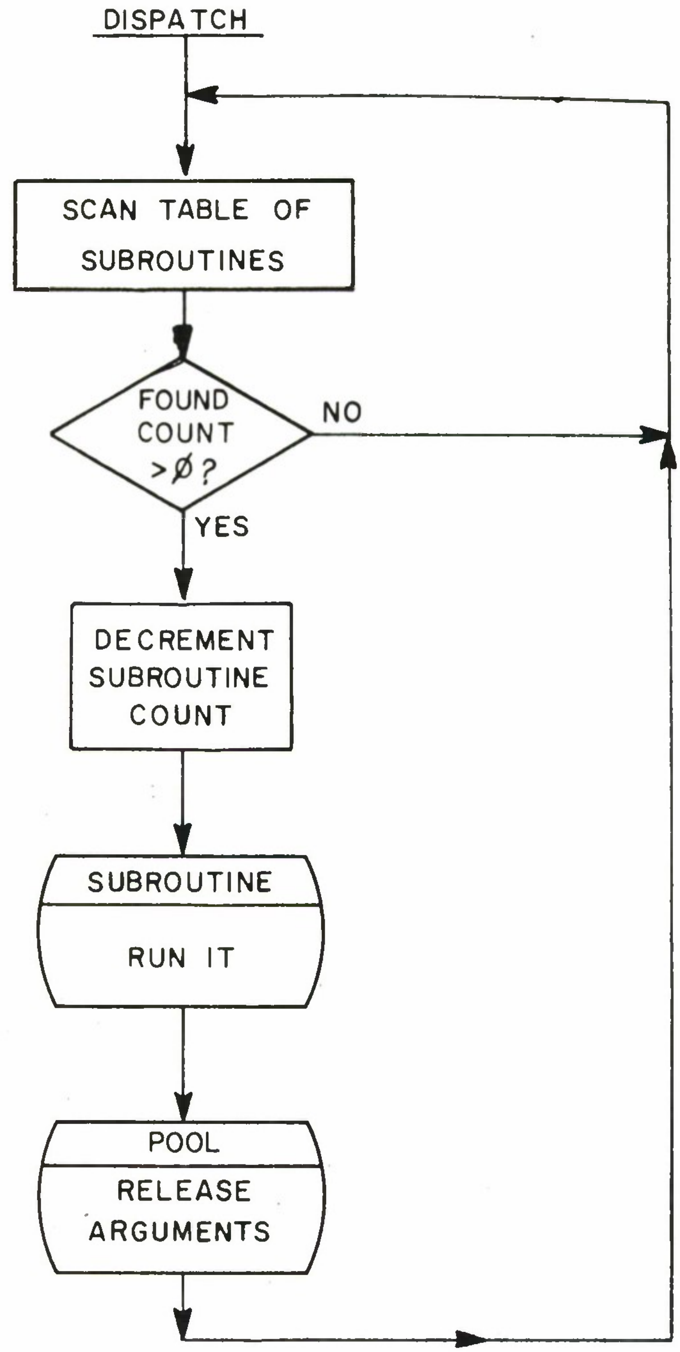
DISPATCH AND SCHEDULER FLOWCHARTS

NOTE:

In accordance with conventional flowcharting techniques, special connector symbols are used to represent an entry from, or an exit to, another part of the program flowchart.

The symbol is used to indicate an exit from a page. The letter within the symbol indicates the entry point that will be found on another page.

The symbol is used to indicate an exit to another part of the program on the same page. It is also used in the following flowcharts to indicate the entry point from another page.



1A-43,570

Figure E-1 DISPATCH

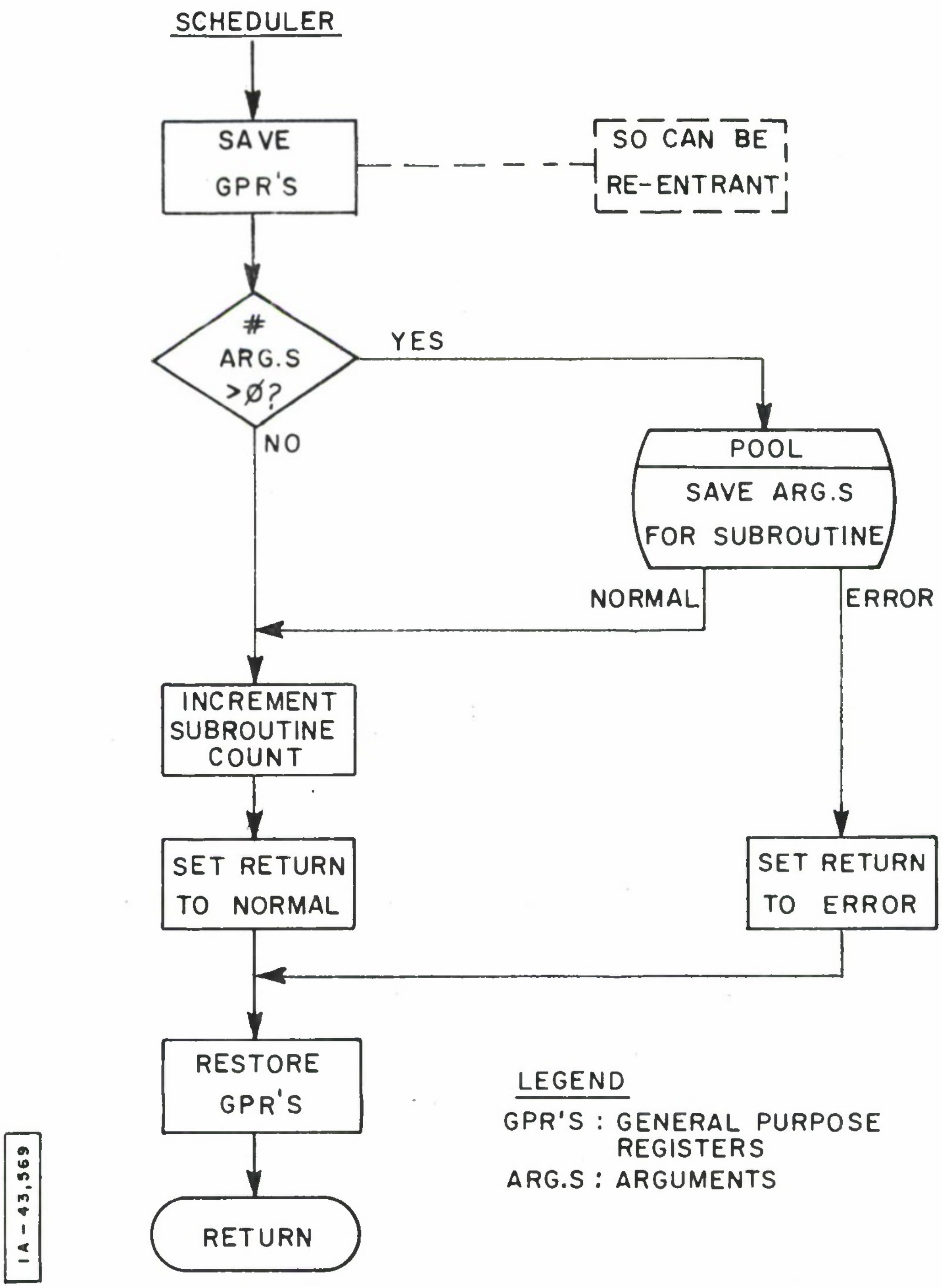


Figure E-2 SCHEDULER

APPENDIX F

POOL HANDLER FLOWCHARTS

NOTE:

In accordance with conventional flowcharting techniques, special connector symbols are used to represent an entry from, or an exit to, another part of the program flowchart.

The symbol is used to indicate an exit from a page. The letter within the symbol indicates the entry point that will be found on another page.

The symbol is used to indicate an exit to another part of the program on the same page. It is also used in the following flowcharts to indicate the entry point from another page.

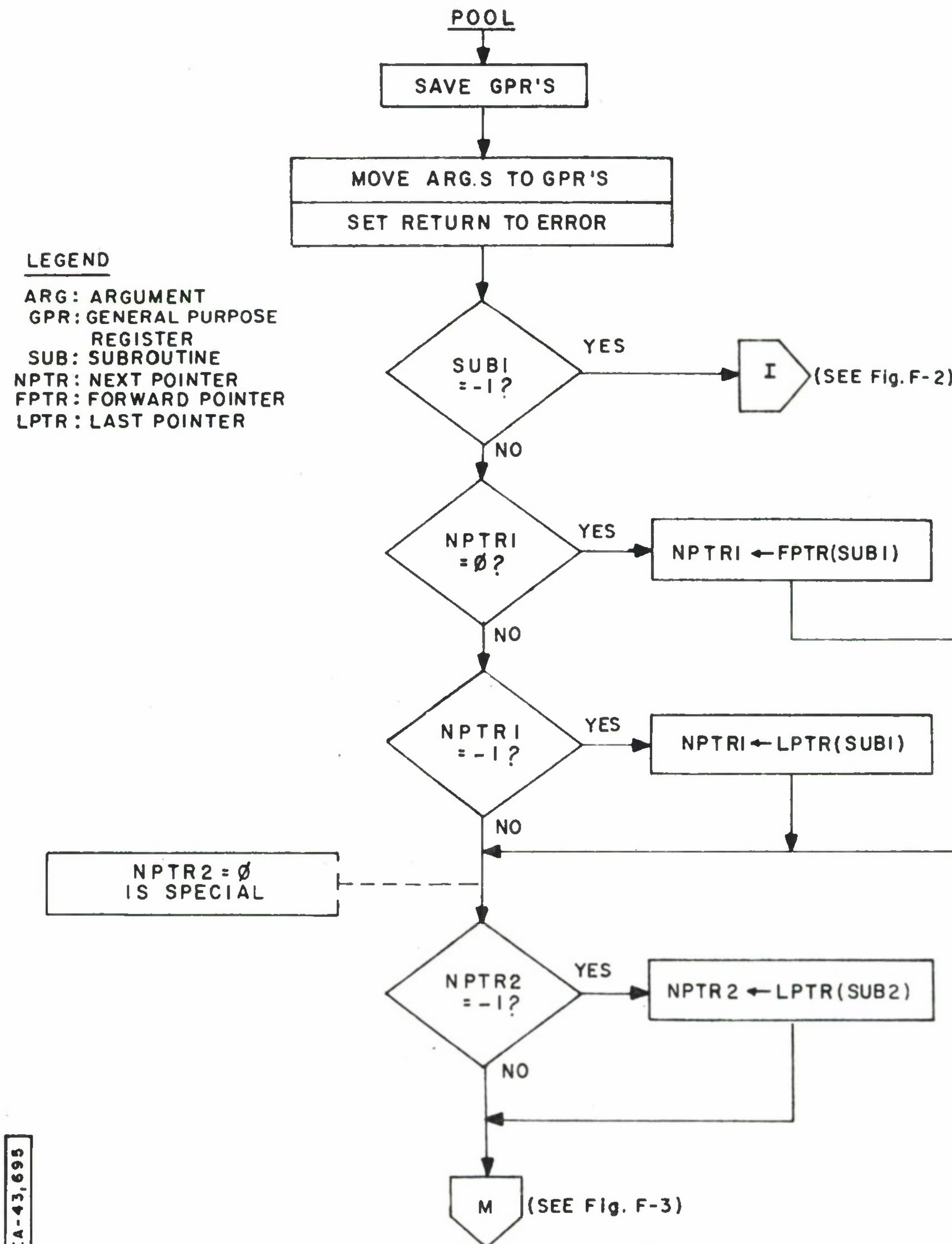


Figure F-1 POOL HANDLER

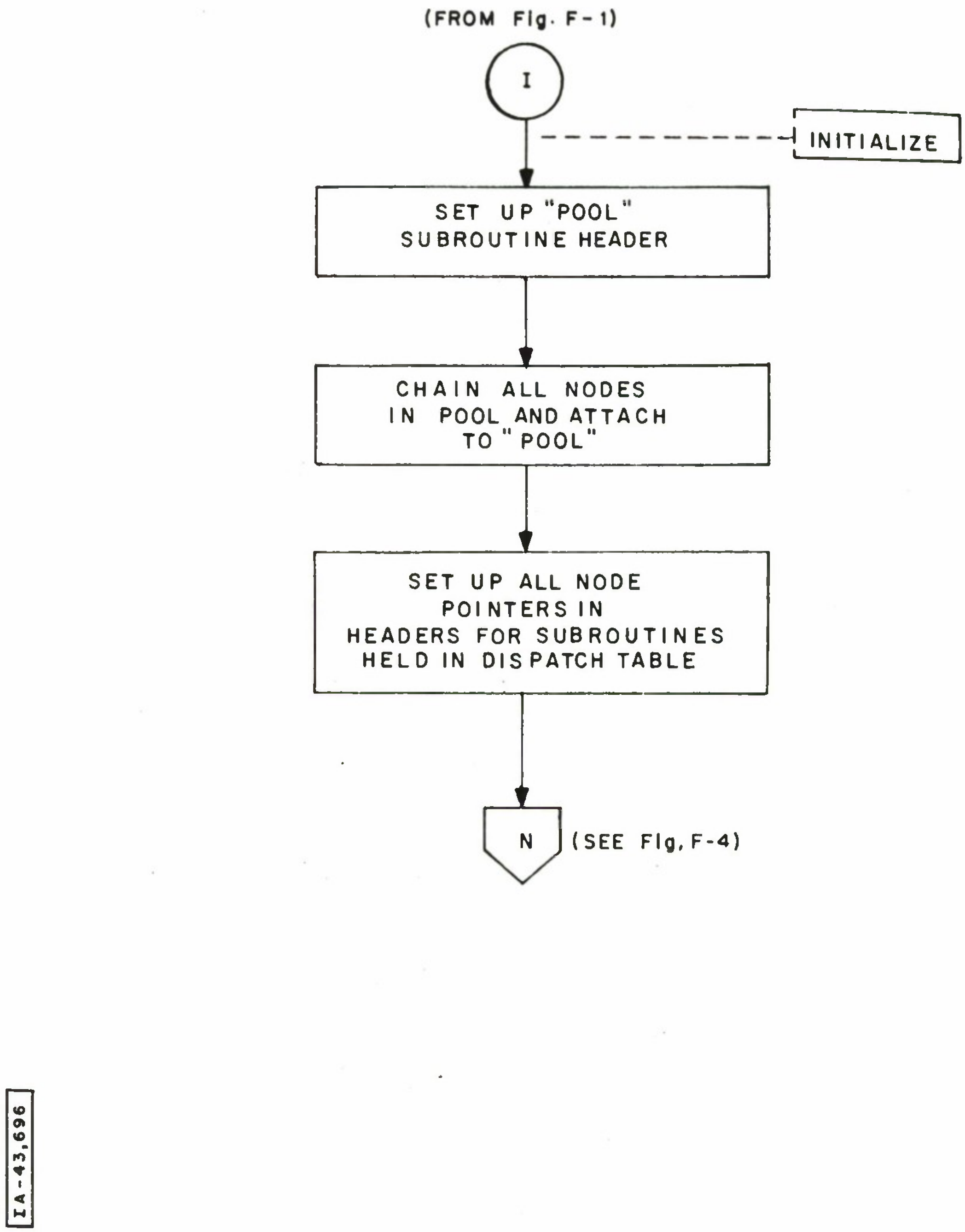


Figure F-2 POOL HANDLER (CONTINUED)

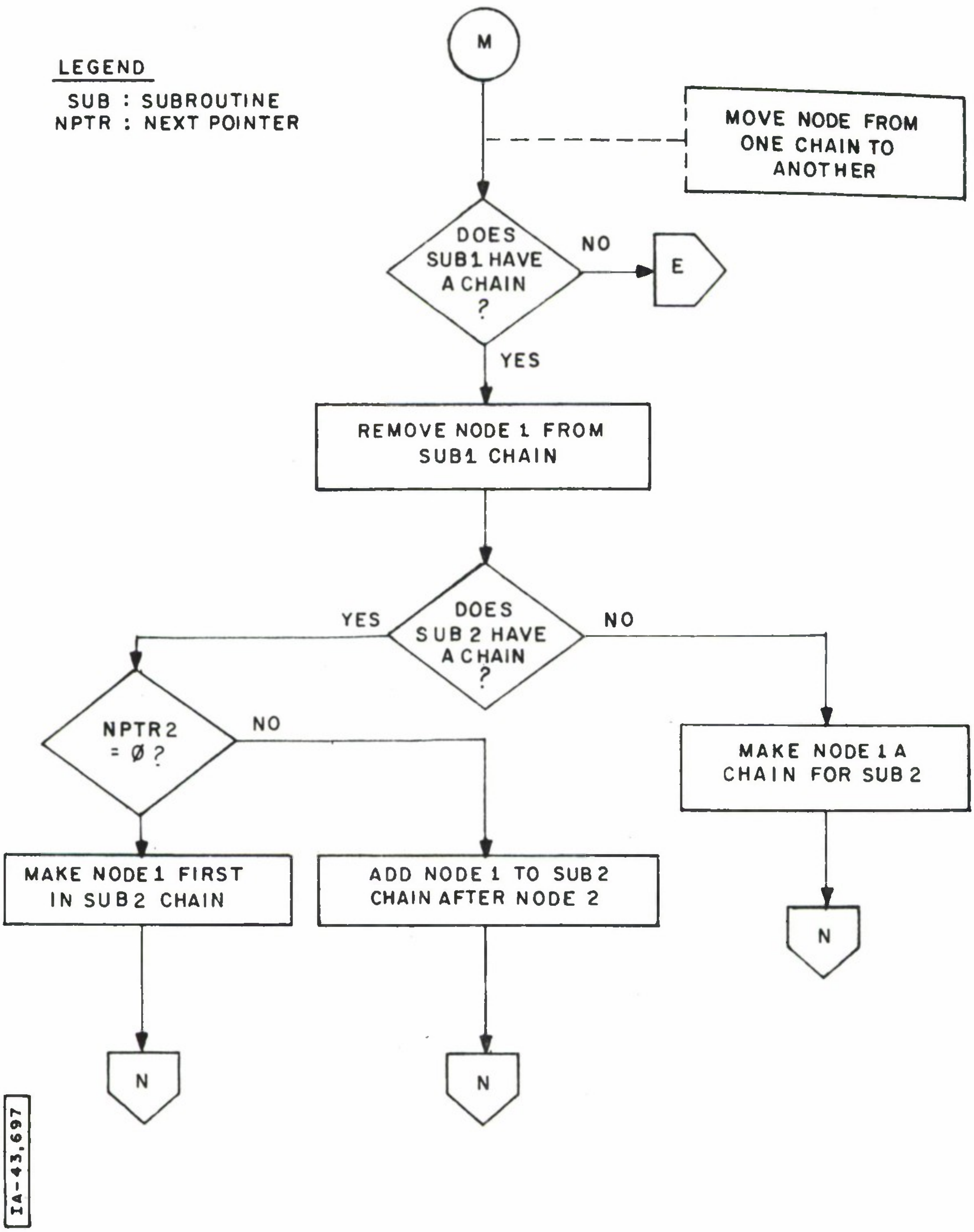
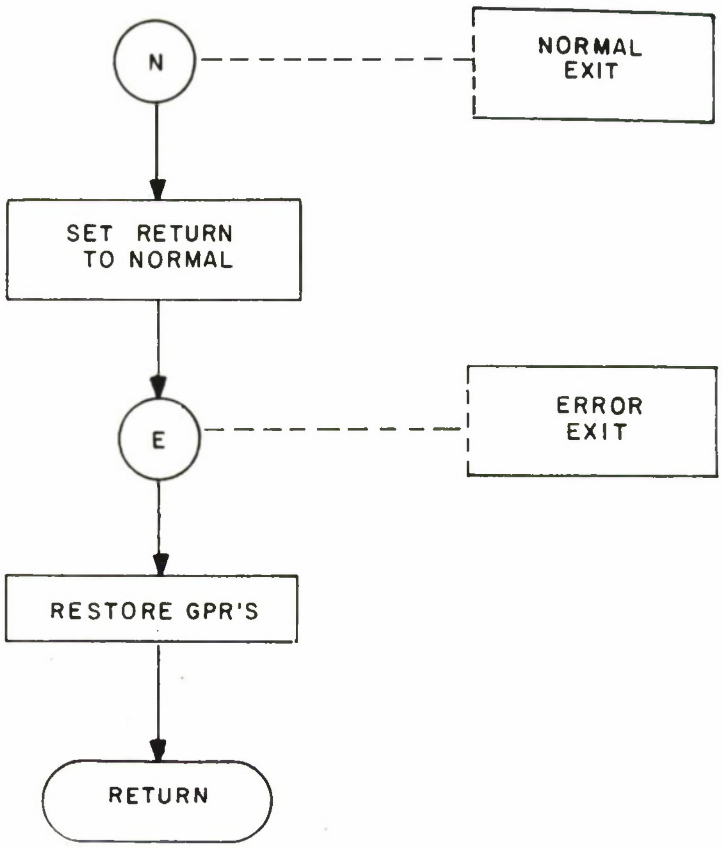


Figure F-3 POOL HANDLER (CONTINUED)

LEGEND

GPR : GENERAL PURPOSE REGISTER



IA-43,698

Figure F-4 POOL HANDLER (CONCLUDED)

REFERENCES

1. W. T. Grinnell, AFBITS Experimental Equipment Evaluations, The MITRE Corporation, Bedford, Mass., ESD-TR-74-313, February 1975.

2. Digital Equipment Corporation, PDP-11 Paper Tape Software Programming Handbook, Chapter 5 & 7, Maynard, Mass., 1973.

3. Digital Equipment Corporation, CAPS 11 User´s Guide to PDP-11, Maynard, Mass., 1973.

4. A. E. Spencer and F. S. Vigilante, "System Organization and Objectives," Bell System Technical Journal, New York, Vol. 48, No 2 ESS, October, 1969, page 2615.

5. Digital Equipment Corporation, PDP-11 Processor Handbook, Appendix C, 1973.

6. W. O. Fleckenstein "Bell System ESS Family - Present and Future," International Switching Symposium Record, Munich, 1974.

7. E. J. Reines, E. G. Platt and M. C. Byckowski," TCS Organization and Objectives," Electrical Communication, Vol. 48, Number 4, London, 1973, page 371.